Міністерство освіти і науки України
Вінницький державний технічний університет

О.М. Роїк, Р.Г. Тадевосян

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

ЧАСТИНА 2

ЕЛЕМЕНТИ ЗАГАЛЬНОЇ АЛГЕБРИ, БУЛЕВІ ФУНКЦІЇ, ТЕОРІЯ ГРАФІВ І КОМБІНАТОРИКА

Затверджено Ученою радою Вінницького державного технічного університету як навчальний посібник для студентів бакалаврського напрямку 6.0804 – "Комп'ютерні науки" спеціальностей 7.080404 – "Інтелектуальні системи прийняття рішень" та 7.080403 – "Програмне забезпечення автоматизованих систем" денної та заочної форм навчання. Протокол № 2 від 28 вересня 2000 р.

Вінниця ВДТУ 2003

УДК 519.85
Р 65

Р е ц е н з е н т и:

В.М. Міхалевич, доктор технічних наук, професор
Н.Р. Кондратенко, кандидат технічних наук, доцент
Л.В.Крупельницький, кандидат технічних наук

Рекомендовано до видання Ученою радою Вінницького державного технічного університету Міністерства освіти і науки України

Роїк О.М., Тадевосян Р.Г.
Р65 **Основи дискретної математики. Ч2. Елементи загальної алгебри, булеві функції, теорія графів і комбінаторика.** Навчальний посібник. - Вінниця: ВДТУ, 2003.- 116с.

У посібнику розглядаються алгебраїчні системи, теорія графів і комбінаторика. В окремих розділах розглядаються алгебра булевих функцій і графи. В останньому розділі розглядаються елементи комбінаторики. Посібник розроблений у відповідності з планом кафедри та програми до дисциплін "Основи дискретної математики".

УДК 519.85

1 ЕЛЕМЕНТИ ЗАГАЛЬНОЇ АЛГЕБРИ

Якщо у минулих століттях і на початку XX століття алгебру вивчало обмежене число алгебраїчних структур, то зараз можна дати загальне означення алгебри – а саме: наука про властивості нескінченностей, на яких визначена та або інша система операцій і відношень. У розвиток такого погляду на алгебру вніс великий вклад А.І. Мальцев. Зокрема, він увів поняття алгебраїчної системи, що і є темою даного розділу. Завдяки його роботам стало ясно, що алгебра і математична логіка – дві тісно пов'язані між собою дисципліни.

1.1 Алгебри, моделі і системи

Означення (Відношення). n-арним (*n-місним*) *відношенням* на множині A називається підмножина *n-ого* декартового степеня A^n множини A.

Означення (Алгебраїчна операція). n-арною (*n-місною*) *алгебраїчною операцією* або просто *операцією*, визначеною на множині A називається n-місна функція $f: A^n \to A$.

Число n для n-арної операції f (n-арного відношеня r) називається *арністю* операції f і позначається як $n(f)$ ($n(r)$). Арність відношеня – це числа, що більші нуля. Арності операцій – це числа, що більші або дорівнюють нулю. Операції нульової арності є функціями з областю визначення, що складаються з одного елемента (*n-ка* нульової довжини) і ототожнюються із самим значенням функції.

Властивості відношень

Рефлексивність. Відношення R *рефлексивне*, якщо $(x, x) \in R$ для будь-якого $x \in A$.

Іррефлексивність. Відношення R *іррефлексивне*, якщо $(x, x) \notin R$ для будь-якого $x \in A$.

Симетричність. Відношення R *симетричне*, якщо $(y, x) \in R$ для будь-яких $x, y \in A$, таких що $(x, y) \in R$.

Антисиметричність. Відношення R *антисиметричне*, якщо $(y, x) \notin R$ для будь-якого $x, y \in A$, таких що $(x, y) \in R$.

Транзитивність. Відношення R *транзитивне*, якщо $(x, z) \in R$ для будь-якого $x, y, z \in A$, таких що $(x, y), (y, z) \in R$.

Відношення еквівалентності. Якщо відношення рефлексивне, симетричне і транзитивне, то його називають *відношенням еквівалентності*.

Відношення порядку. Якщо відношення рефлексивне, антисиметричне і транзитивне, то його називають *відношенням порядку*.

Властивості операцій

Ідемпотентність. Операція * *ідемпотентна*, якщо $x * x = x$ для будь-якого $x \in A$.

Коммутативність. Операція * *коммутативна*, якщо $x * y = y * x$ для будь-яких $x, y \in A$.

Антикомутативність. Операція * *антикоммутативна*, якщо $x * y \neq y * x$ для будь-яких $x, y \in A$.

Ассоціативність. Операція * *ассоціативна*, якщо $x * (y * z) = (x * y) * z$ для будь-яких $x, y, z \in A$.

Дистрибутивність. Операція * *дистрибутивна* щодо операції $\circ$, якщо $x \circ (y * z) = (x \circ y) * (x \circ z)$ для будь-яких $x, y, z \in A$.

Взаємно-обернені операції. Операції * і $\circ$ називають взаємно-оберненими, якщо $x * y = z$ тільки тоді, коли $z \circ y = x$ для будь-яких $x, y, z \in A$.

Нейтральний елемент. Про операцію * говорять, що вона *має нейтральний елемент*, якщо в множині A існує елемент (позначимо його e), такий що $x * e = x$ для будь-якого $x \in A$. Якщо розглянута операція позначається знаком +, то нейтральний елемент зазвичай називають *нулем*, а якщо знаком * (помножити), то – *одиницею*.

Обернений елемент. Про операцію * з нейтральним елементом e говорять, що для неї елемент $x \in A$ має обернений елемент, якщо для нього в множині A можна знайти елемент (позначимо його x'), такий що $x * x' = e$. Якщо для всіх елементів існують обернені елементи, то операцію називають *оберненою*.

Якщо деяка властивість не виконується (тобто для деяких елементів умова порушується), то до назви властивості додають "не". Так, наприклад, якщо говорять про нерефлексивне відношення, це означає, що є елемент, для якого порушується властивість рефлексивності. Якщо множина A скінченна, алгебраїчну операцію на ній можна визначити у вигляді таблиці.

Приклад. Таблиця операції (+ mod 5) на множині {0, 1, 2, 3, 4}.

(+mod 5)	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Крім того, що таблиця дає визначення операції, вона наочно описує деякі властивості операції. Зокрема, комутативність операції відповідає симетричності таблиці щодо головної діагоналі.

Означення (Алгебра). *Алгеброю* називається об'єкт $<A; \Omega_F>$, що складається з непорожньої замкненої множини A із заданою на ній множиною алгебраїчних операцій Ω_F.

Означення (Алгебраїчна модель). *Алгебраїчною моделлю* називається об'єкт $<A; \Omega_R>$, що складається з непорожньої замкненої множини A із заданою на ній множиною множини відношень Ω_R.

Означення (Алгебраїчна система). *Алгебраїчною системою* називається об'єкт $<A; \Omega_F; \Omega_R>$, що складається з трьох множин:

- непорожньої замкненої множини A;
- множини алгебраїчних операцій Ω_F, що визначені на множині A;
- множини відношень Ω_R, визначених на множині A.

Означення (Замкнена множина). Множина A називається *замкненою* щодо операцій коли значення результатів операцій на аргументах із A належать множині A.

Означення (Носії і сигнатури). Множина A називається *носієм* відповідної алгебри, алгебраїчної моделі і алгебраїчної системи, для яких множини алгебраїчних операцій і відношень, а також визначення арностей складають *сигнатури* відповідних алгебр моделей і систем.

Означення (Типи алгебр, моделей і систем). *Типом* алгебр та алгебраїчних моделей і систем називаються відповідно вектор арностей операцій, вектор арностей відношень і пари векторів арностей операцій і відношень.

Приклади.

1. Об'єкт $\{R; +\ , \cdot\ \}$ є алгеброю і називається *полем дійсних чисел*. Обидві операції є бінарними і тому тип цієї алгебри (2,2).

2. Нехай $N_p = \{0,1,2, \dots , p\text{-}1\}$. Визначимо на N_p операції $\oplus$ (додавання за модулем p) і $\otimes$ (множення за модулем p) таким чином: $a\oplus b=c$, $a\otimes b=d$, де c і d - залишки від ділення на p чисел $a+b$ і $a\cdot b$ відповідно. Наприклад, якщо p=7, то N_p={0,1,...,6}, $3\oplus4=0$, $3\otimes4=5$, $4\oplus6=3$. Часто операції $\oplus$ і $\otimes$ позначають як $a+b=c\ (mod\ p)$, $ab=d\ (mod\ p)$. Якщо p - просте число, то $\{N_p, \oplus, \otimes\}$ є алгеброю, що називається *скінченим полем характеристики p.*

3. Об'єкт $\{\mathbf{N}; +, \cdot, <\}$ є алгебраїчною системою типу < 2, 2, 2 >, оскільки обидві її операції і відношення визначені для будь-яких двох натуральних чисел, при цьому результат операції знову таки буде натуральним числом.

4. Об'єкт $\{\mathbf{N}; +, \bar{}\ ,<\}$ не є алгебраїчною системою, оскільки результат віднімання натуральних чисел не завжди буде натуральним числом.

Нижче, оскільки поняття алгебраїчної системи узагальнює поняття алгебр і моделей, будемо оперувати поняттям алгебраїчних систем, причому таких, що містять скінченне число операцій і відношень.

1.2 Ізоморфізм алгебраїчних систем

Стандартний алгебраїчний підхід до розгляду алгебр, моделей і систем є відволіканням від властивостей елементів носія, що не пов'язані з операціями і відношеннями і від способів визначення (обчислення) операцій і відношень. Розглядаються тільки їх властивості у рамках алгебр, моделей і систем. Очевидно, що алгебри, моделі і системи з різними типами мають істотно різну побудову. Коли ж вони мають однаковий тип, то наявність у них подібності характеризується за допомогою понять гомоморфізму та ізоморфізму.

Означення (Гомоморфізм). Нехай $A = <A; f_1,...,f_k; r_1,...,r_l>$ і $B = <B; g_1,...,g_k; p_1,...,p_l>$ – дві алгебраїчні системи одного типу $<m_1,...,m_k; n_1,...,n_l>$. Відображення $\varphi : A \to B$ називається *гомоморфізмом* алгебраїчної системи A у B, якщо виконуються такі умови:

$$\begin{aligned}&- \varphi(f_i(x_1,...,x_{mi})) = g_i(\varphi(x_1),...,\varphi(x_{mi})),\\ &- (x_1,...,x_{nj}) \in r_j \Rightarrow (\varphi(x_1),...,\varphi(x_{mi})) \in p_j,\\ &(\forall x_1, x_2, ... \in A,\ \forall i : 1 \le i \le k, \forall j : 1 \le j \le l).\end{aligned}$$

Сутність даної умови полягає в тому, що незалежно від того, чи виконана спочатку операція f_i у A, а потім здійснено відображення φ, або спочатку здійснено відображення φ, а потім виконана операція g_i в B , результат буде однаковий.

Приклад (Гомоморфізм). Будь-яке відображення будь-якої моделі $<A; p>$ типу $< 2 >$ на модель $< A; \varsigma >$ (де ς – порожнє бінарне відношення) є гомоморфізмом, оскільки перша умова виконується через відсутність операцій, а друга – через те, що посилка імплікації завжди помилкова.

Означення (Ізоморфізм). Якщо гомоморфізм є бієкцією і обернене відображення теж є гомоморфізмом, то такий гомоморфізм називається *ізоморфізмом*. Алгебраїчні системи, для яких існує ізоморфізм, називаються *ізоморфними*. Інакше кажучи, ізоморфізм алгебраїчних систем $A = <A; f_1,...f_k; r_1,...,r_l>$ і $B = <B; g_1,...,g_k; p_1,...,p_l>$ одного типу $<m_1,...,m_k; n_1,...,n_l>$ – це взаємооднозначне відображення φ множини A на B, причому таке, що виконуються умови:

$$\begin{aligned}&- \varphi(f_i(x_1,...,x_{mi})) = g_i(\varphi(x_1),...,\varphi(x_{mi})),\\ &- (x_1,...,x_{nj}) \in r_j \Leftrightarrow (\varphi(x_1),...,\varphi(x_{nj})) \in p_j,\\ &(\forall x_1, x_2, ... \in A,\ \forall i : 1 \le i \le k, \forall j : 1 \le j \le l).\end{aligned}$$

Для алгебр друга умова виконується автоматично, оскільки для алгебр ізоморфізми – це гомоморфізми, що є бієкцією.

Приклад (Ізоморфізм алгебр). Покажемо, що алгебра $< \mathbf{R}; + >$ буде ізоморфна алгебрі $<\mathbf{R}_+ ; \cdot >$. Визначимо відображення $\varphi : \mathbf{R} \to \mathbf{R}_+$ як $\varphi(x)$

$= e^x$. Це відображення є бієкцією і $\varphi(x+y) = e^{(x+y)} = e^x \cdot e^y = \varphi(x) \cdot \varphi(y)$.

Приклад (Ізоморфізм моделей). Покажемо, що моделі $<\mathbf{R}; \leq>$ і $<\mathbf{R}; \geq>$ ізоморфні. Визначимо відображення $\varphi(x) = -x$. Це відображення є бієкцією і $\varphi(x) \geq \varphi(y) \Leftrightarrow -x \geq -y \Leftrightarrow x \leq y$.

Означення (Автоморфізм). Ізоморфізм алгебраїчної системи на собі називається *автоморфізмом*. Автоморфізм, що є тотожним відображенням, називається *тривіальним*.

Задача. Довести, що алгебри $<\mathbf{Q}; +>$ і $<\mathbf{Z}; +>$ не ізоморфні.

Розв'язання. Потужності множин $\boldsymbol{Q}$ і $\boldsymbol{Z}$ рівні. Значить існує бієктивне відображення з $\boldsymbol{Q}$ на $\boldsymbol{Z}$. Питання в тому, чи можна знайти таке бієктивне відображення, що зберігало б операцію +. Багато властивостей операції + в $<\boldsymbol{Q}; +>$ і в $<\boldsymbol{Z}; +>$ збігаються. Дійсно вони обидві комутативні, асоціативні, мають нейтральний елемент і т.д. Однак властивості операції + в $<\boldsymbol{Z}; +>$, зв'язані з 1, не виконуються в $<\boldsymbol{Q}; +>$. Це дає можливість довести відсутність ізоморфізму. Припустимо, що ізоморфізм φ алгебри $<\boldsymbol{Q}; +>$ на $<\boldsymbol{Z}; +>$ існує. Нехай p/q – прообраз 1 при відображенні φ, тобто $\varphi(p/q)=1$. Нехай $\varphi(p/2q)=n$. Однак оскільки φ – ізоморфізм, то $\varphi(p/2q+p/2q) = \varphi(p/2q)+\varphi(p/2q)$. Одержуємо, що $1 = n + n$, де n – деяке ціле число. Однак, як відомо, таких цілих чисел немає. Таким чином приходимо до протиріччя, що і доводить відсутність ізоморфізму.

Задача. Знайти кількість автоморфізмів моделі $<\{2,3,4,5,6,7\}; r>$, де r – відношення взаємної непростоти.

Розв'язання. У відношенні r знаходяться пари чисел (2,4), (2,6), (3,6), (4,6) (у такому й у зворотному порядку) і пари виду (x, x). Тільки одне число (6) знаходиться у відношенні r з трьома іншими числами й одне число (3) рівне з одним. Значить образ числа (6) при автоморфізмі повинен бути знову (6), а образ (3) – (3). Таким чином $\varphi(3)=3$, $\varphi(6)=6$. Образ числа (5) може дорівнювати п'яти або семи. Так само два і чотири можуть відображатися на себе й один на одного, оскільки з погляду відношення r вони симетричні – знаходяться у відношенні один з одним і з (6). Разом одержуємо $2 \cdot 2 = 4$ варіанти автоморфізму φ. Випишемо їх у таблицю:

I	2	3	4	5	6	7
φ_1	2	3	4	5	6	7
φ_2	4	3	2	5	6	7
φ_3	2	3	4	7	6	5
φ_4	4	3	2	7	6	5

1.3 Підсистеми алгебраїчних систем

Означення (Підсистема). *Підсистемою* алгебраїчної системи $<A;\Omega_F;\Omega_R>$ називається алгебраїчна система $<A';\Omega_F';\Omega_R'>$, в якій: $A' \subseteq A$; значення всіх операцій з Ω_F' на A' збігаються з Ω_F; відношення з Ω_R' на A' збігаються з відношеннями з Ω_R. При цьому множина A' є *замкненою* у системі $<A; \Omega_F; \Omega_R>$. Відзначимо, що гомоморфні образи алгебр завжди ізоморфні підалгебрам, однак гомоморфні образи моделей – не обов'язково є ізоморфними підмоделям даної моделі.

Приклад (Гомоморфні образи моделі). Модель $<\{0,1\}; <>$ можна гомоморфно відобразити на моделях $<\{0,1\}; \leq>$, $<\{0\}; \{(0,0)\}>$ і т.п., однак вони не є підмоделями вихідної моделі.

Теорема (Переріз підсистем). Переріз довільної сукупності підсистем будь-якої алгебраїчної системи або порожній, або є підсистемою.

Доведення. Нам треба довести, що переріз підсистем замкнений щодо операцій алгебраїчної системи. Нехай f – n-місна операція алгебраїчної системи, а $a_1,...,a_n$ – елементи перерізу. Оскільки операція f замкнена на кожній підсистемі, значить елемент $f(a_1,...,a_n)$ належить усім підсистемам, а значить і їхньому перерізу. Таким чином операція замкнена.

Приклад (Переріз алгебр). Алгебри $<\{0, 2, 4\}; (+ \bmod 6)>$ і $<\{0, 3\}; (+ \bmod 6)>$ є підалгебрами алгебри $<\{0, 1, 2, 3, 4, 5\}; (+ \bmod 6)>$. Перерізом цих алгебр буде алгебра $<\{0\}; +>$ з носієм, що складається з одного елемента - 0 і операцією, яку ми позначили як +, оскільки вона може бути застосована тільки до елемента 0 і збігається з результатом звичайного додавання.

Означення (Замикання множини в алгебраїчній системі). Носій мінімальної алгебраїчної системи, що містить множину A, називається *замиканням множини A в алгебраїчній системі.*

Приклад (Замикання множини). Замиканням множини $\{-1,1\}$ в алгебрі $<\mathbf{Q}; +>$ є множина $\mathbf{Z}$ цілих чисел, оскільки $<\mathbf{Z}; +>$ є алгеброю, і, якщо підалгебра алгебри $<\mathbf{Q}; +>$ містить -1 і 1, вона містить усі цілі числа.

Алгебраїчна система може бути ізоморфна своїй підсистемі.

Приклад (Підалгебра, ізоморфна алгебрі). Покажемо, що алгебри $<\mathbf{N};+>$ і $<\{2,4,6,...\};+>$ ізоморфні. Визначимо відображення φ: $\mathbf{N} \to \{2,4,6,...\}$ як $\varphi(x) = 2x$. Це відображення є бієкцією і $\varphi(x+y) = 2(x+y) = 2x + 2y = \varphi(x) + \varphi(y)$.

1.4 Прямий добуток алгебраїчних систем

Означення (Прямий добуток систем). *Прямим добутком* алгебраїчних систем $A = <A; f_1,..., f_k; r_1,...,r_l>$ і $B = <B; g_1,...,g_k; p_1,...,p_l>$ типу

$<m_1,...,m_k;\ n_1,...,n_l>$ називається алгебраїчна система $A \times B = <A \times B;\ h_1,...,h_k; q_1,...,q_l>$ того ж типу, така що

$$h_i((x_1,y_1),...,(x_{mi}, y_{mi})) = (f_i(x_1,...,x_{mi}),\ g_i(y_1,...,y_{mi}))$$

$$((x_1,y_1),...,(x_{mj},y_{mj})) \in q_j \Leftrightarrow (x_1,...,x_{mj}) \in r_j\ i\ (y_1,...,y_{mj}) \in p_j$$

для будь-яких $x_1, x_2,... \in A$, $y_1, y_2,... \in B$, для будь-яких $i : 1 \le i \le k$, для будь-яких $j : 1 \le j \le l$.

Пряме множення алгебраїчної системи A саму на себе n раз називається *степенем* алгебраїчної системи A і позначається як A^n.

Приклад (Прямий добуток). Розглянемо прямий добуток алгебраїчної системи $A = <\mathbf{R};\ +;\ \le>$ на собі. Носієм алгебраїчної системи A^2 є множина пари чисел (x, y) з операцією покоординатного додавання і відношенням порядку $\le$, таким що $(x_1, y_1) \le (x_2, y_2) \Leftrightarrow x_1 \le x_2$ і $y_1 \le y_2$. Тобто одна пара менше або дорівнює іншій тільки тоді, коли кожна координата першої пари менша або дорівнює відповідній координаті другої пари.

1.5 Приклади алгебраїчних систем

Числа з додаванням і множенням. Покажемо, що дійсні числа з додаванням і множенням є алгеброю. Дійсно, і сума і добуток визначені для будь-яких двох дійсних чисел і є знову дійсними числами. Таким чином, додавання і множення дійсних чисел – алгебраїчні операції і $<\mathbf{R};\ +, \cdot>$ – алгебра. Крім того, раціональні (цілі, цілі додатні і натуральні) числа з додаванням і множенням теж утворять алгебру. Дійсно, сума і добуток двох раціональних чисел знову є раціональними, відповідно ціле, ціле додатне і натуральне число. Таким чином, множини раціональних є замкнутими в $<\mathbf{R};\ +, \cdot>$, при цьому $<\mathbf{Q};\ +, \cdot>$, $<\mathbf{Z};\ +, \cdot>$, $<\mathbf{Z}_-;\ +, \cdot>$, $<\mathbf{N};\ +, \cdot>$ – алгебри, що є підалгебрами $<\mathbf{R};\ +, \cdot>$.

Вектори на площині. Множина усіх векторів на площині з операцією додавання утворює алгебру $<\mathbf{R}^2;\ +>$. Покажемо, що її можна розглядати як квадрат алгебри $<\mathbf{R};\ +>$ дійсних чисел з додаванням. Дійсно, сума двох векторів (x_1, y_1) і (x_2, y_2) визначається покоординатно, тобто як $(x_1, y_1) + (x_2, y_2) = (x_1+x_2, y_1+y_2)$ для будь-яких $x_1, x_2, y_1, y_2 \in \mathbf{R}$.

Алгебра підмножин. Одна з алгебраїчних систем, з якими ми часто зіштовхуємося в математиці є *алгеброю підмножин*. Розглянемо всі підмножини деякої множини A, яку позначимо як $P(A)$. Операції об'єднання, перерізу двох множин і доповнення множини X до множини A є алгебраїчними операціями. Дійсно, вони визначені для будь-яких підмножин множини A, а результати цих операцій є підмножинами множини A.

Порожня множина і сама множина A теж є підмножинами A. Отже, алгеброю підмножин множини A називається алгебра $<P(A);\ \cup, \cap, \neg\ \varnothing, A>$.

1.6 Класи алгебраїчних систем

Традиційним способом визначення, вивчення і класифікації алгебраїчних систем є аксіоматичний спосіб. Алгебра вивчає не окремі алгебраїчні системи, а класи алгебраїчних систем, а саме ті алгебраїчні системи, що задовольняють деяку систему аксіом. Тобто вивчаються в основному властивості, що є загальними для всіх алгебраїчних систем з даного класу замість того, щоб вивчати властивості кожної з них окремо. Такий підхід виявляється не тільки продуктивним, але й найбільш природним. Часто для класів алгебраїчних систем розглядаються більш вузькі класи, що отримують додаванням нових аксіом. У наступних розділах розглядаються такі класи алгебраїчних систем як частково-впорядковані множини, грати, булеві алгебри, графи, групи і напівгрупи.

Графи

Часто буває незручно застосовувати поняття алгебраїчної системи з однорідним носієм і розглядають *багатоосновні* алгебраїчні системи $<A_1,..A_n; \Omega_F; \Omega_R>$, де: $A_1,..., A_n$ – не порожні множини, що перетинаються; Ω_F – множина операцій, що визначена на декартових добутках множин $A_1,..., A_n$; Ω_R – множина відношень між елементами множин $A_1,..., A_n$.

Означення (Граф). *Графом* називається двоосновна модель $<V, E; i>$, де i – бінарне відношення множин V і E, таке, що кожен елемент $e \in E$ знаходиться у відношенні i або з одним, або з двома елементами множин у V. При цьому елементи множини V називаються *вершинами* графа, елементи множини E називаються *ребрами* графа, а відношення i – відношенням *інцидентності*. Вершини, що інциденті одному ребру, називаються *суміжними*. Графи зображуються у вигляді геометричних фігур, де вершини графа зображуються точками, а ребра – лініями, що з'єднують точки відповідних вершин для яких ці ребра інциденті.

Означення (Петлі, кратні ребра). Якщо ребро інцидентне тільки одній вершині, його називають *петлею*. Ребра називаються *кратними*, якщо вони інциденті тим самим вершинам. Граф без кратних ребер можна визначити як модель $<V; c>$, де c – бінарне відношення суміжності.

Правило. Для будь-яких двох графів $G_1=<V_1, E_1; i_1>$ і $G_2=<V_2, E_2; i_2>$ без кратних ребер вірним є таке: якщо моделі $<V_1; c_1>$ і $<V_2; c_2>$, де c_1 (c_2) є відношенням суміжності вершин у графі G_1 (G_2), – ізоморфні, то і самі графи G_1 і G_2 ізоморфні.

Означення (Простий граф). Граф без кратних ребер і петель називається *простим графом*.

Частково-впорядковані множини

Означення (Частково-впорядковані множини). Частково-впорядкованою множиною називається модель $<A; \leq>$, бінарне відношення $\leq$

якої задовольняє таку систему аксіом:

1. $\forall a \in A$ виконується $(a, a) \in \leq$ або $(a \leq a)$;

2. $\forall a, b \in A$, з $a \leq b$, $b \leq a$ випливає $a = b$;

3. $\forall a, b, c \in A$, таких що $a \leq b$, $b \leq c$ виконується $a \leq b$.

Означення (Гратчасто-упорядкована множина). Множина $<A; \leq>$ називається *гратчасто-впорядкованою*, якщо для будь-яких двох елементів a, $b \in A$ існує точна нижня грань $\inf(a,b) \in A$ і точна верхня грань $\sup(a,b) \in A$.

Гратки

Означення (Гратки). *Гратками* називається алгебра $<A; \vee, \wedge>$, операції $\vee$ і $\wedge$ якої задовольняють такій системі аксіом:

1. $\forall a \in A$ виконуються тотожності:
   $a \vee a = a, a \wedge a = a$;
2. $\forall a, b \in A$ виконуються тотожності:
   $a \vee b = b \vee a, a \wedge b = b \wedge a$;
3. $\forall a, b, c \in A$ виконуються тотожності:
   $a \vee (b \vee c) = (a \vee b) \vee c, a \wedge (b \wedge c) = (a \wedge b) \wedge c$;
4. $\forall a, b \in A$ виконуються тотожності:
   $a \vee (a \wedge b) = a, a \wedge (a \vee b) = a$ (закони поглинання).

Теорема (Зв'язок частково-впорядкованої множини і граток). Якщо алгебра $<A; \vee, \wedge>$ є гратками, то модель $<A; \leq>$, де відношення $\leq$ визначене як $x \leq y \Leftrightarrow x \wedge y = x$ є гратчасто-впорядкованою множиною. Якщо модель $<A; \leq>$ є гратчасто-впорядкованою множиною, то алгебра $<A; \vee, \wedge>$, де операції визначені як $x \vee y = \sup(x, y)$ *і* $x \wedge y = \inf(x, y)$, є гратками.

Відзначимо, що для гомоморфізмів, підсистем граток і частково-впорядкованих множин ця відповідність не зберігається. Дійсно, підалгеброю будь-яких граток знову таки є гратки, а підмоделлю гратчасто-впорядкованої множини не обов'язково є гратчасто-впорядкована множина.

Приклад (підсистеми частково-впорядкованої множини і граток). Для гратчасто-впорядкованої множини $<\{1,2,3,6\}; \mid>$, де $\mid$ – відношення ділимості, модель $<\{1,2,3\}; \mid>$ є підмоделлю, однак $<\{1,2,3\}$; НЗК, НЗД $>$ не є підграткою гратки $<\{1,2,3,6\}$; НЗК, НЗД $>$. Дійсно, $<\{1,2,3\}$; НЗК, НЗД $>$ – взагалі не алгебра, оскільки НЗК (найменше загальне кратне) не є алгебраїчною операцією на множині $\{1,2,3\}$.

Визначення (Дистрибутивні гратки). Гратки $< A; \vee, \wedge >$, в якій операції $\vee$ і $\wedge$ дистрибутивні одна щодо іншої, тобто для будь-яких елементів $a,b,c \in A$ виконуються такі тотожності:

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c);$$

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c),$$

називаються дистрибутивними гратками.

Визначення (Нуль, одиниця). Елемент $x \in A$ називається *нулем* гратки $<A; \vee, \wedge>$, якщо $\forall\ a \in A$ виконується тотожність $x \vee a = a$. Елемент $x \in A$ називається *одиницею*, якщо $\forall a \in A$ виконується тотожність $x \wedge a = a$.

Твердження (Одиничність нуля й одиниці). Нуль і одиниця єдині, якщо вони існують.

Доведення. Доведемо одиничність нуля (аналогічно доводиться одиничність одиниці). Нехай x і y є нулями. Тоді $x \vee a = a$ для будь-якого елемента $a \in A$ і $y \vee a = a$ для будь-якого елемента $a \in A$, $x \vee y = y$, $y \vee x = x$, однак $x \vee y = y \vee x$. Значить $x = y$.

У силу даного твердження ми можемо позначати нуль і одиницю спеціальними символами, а саме як 0 і 1.

Твердження (Властивості нуля й одиниці). Для будь-якого елемента $a \in A$ виконуються тотожності $0 \wedge a = 0$, $1 \vee a = 1$.

Доведення. Доведемо тотожність $0 \wedge a = 0$ (тотожність $1 \vee a = 1$ доводиться аналогічно). $0 \wedge a = 0 \wedge (0 \vee a)$. А $0 \wedge (0 \vee a)$ за законом поглинання дорівнює 0.

Означення (Доповнення). Елемент a' називається *доповненням* елемента a у гратках $<A; \vee, \wedge>$, якщо $a \vee a' = 1$, $a \wedge a' = 0$.

Означення (Гратки з доповненням). Гратки $<A; \vee, \wedge>$, в яких для кожного елемента $a \in A$ існує доповнення $a' \in A$, називаються *гратками з доповненням*.

Означення (Булеві гратки). Дистрибутивні гратки з доповненням називаються *булевими гратками*.

Приклад (Булеві гратки). Розглянемо всі підалгебри деякої алгебри із скінченним носієм і операціями перерізу двох підалгебр і визначення мінімальної алгебри, що містить дві дані алгебри. В загальному випадку такий об'єкт не є граткою, оскільки переріз підалгебр може бути порожнім. Однак, якщо до підалгебри ми додамо порожню множину, то отримаємо гратки.

Булеві алгебри

Означення (Булева алгебра). *Булевою алгеброю* називається алгебраїчна система $<B; +, \cdot, \neg, 0, 1>$ з бінарними операціями $+$, $\cdot$, унарною операцією $\neg$ і константами 0, 1, що задовольняють таку систему аксіом:
- операції $+, \cdot$ ідемпотентні;
- операції $+, \cdot$ комутативні;
- операції $+, \cdot$ асоціативні;

- операції +, · дистрибутивні одна щодо іншої;

Для будь-яких елементів a, b B виконуються такі тотожності:

- $\neg\neg a = a$;
- $\neg(a + b) = \neg a \cdot \neg b$;
- $a + 0 = a$, $a \cdot 1 = a$;
- $a + \neg a = 1$, $a \cdot \neg a = 0$;
- $a + 1 = 1$, $a \cdot 0 = 0$.

Теорема (Зв'язок булевих граток і булевих алгебр)
Якщо алгебра $<B; +, \cdot, \neg, 0, 1>$ є булевою алгеброю, то $<B; +, \cdot>$ - булева гратка. Якщо алгебра $<B; \vee, \wedge>$ – булева гратка, то $<B; \vee, \wedge, \neg, 0, 1>$, де операція $\neg$ визначена як доповнення булевих граток, 0 і 1 – відповідно як нуль і одиниця в булевих гратках, є булевою алгеброю.

Означення (Атом). На елементах булевої алгебри введемо порядок $\leq$: $x \leq y \Leftrightarrow x \cdot y = x$. *Атомами* назвемо найменші ненульові елементи булевої алгебри щодо цього порядку.

Таким чином, булеві алгебри цілком зводяться до алгебр підмножин.

Приклад. (Гратка, що не зводиться до алгебри підмножин)
Точніше наведемо приклад граток, що не ізоморфні ніякій алгебрі $<A; \cup, \cap>$. Гратки $<\{a, b, c, d, e\}; \vee, \wedge>$ задані таблицями:

∨	a	b	c	d	e
a	a	b	c	d	e
b	b	e	e	e	e
c	c	e	c	e	e
d	d	e	e	d	e
e	e	e	e	e	e

∧	a	b	c	d	e
a	a	a	a	a	a
b	a	b	a	a	b
c	a	a	c	a	c
d	a	a	a	d	d
e	a	b	c	d	e

Частковий порядок, що відповідає даним граткам ($x \leq y \Leftrightarrow x \wedge y = x$), зобразимо у вигляді графа. Розглянемо вирази $(b \wedge c) \vee d$ і $(b \vee d) \wedge (b \vee c)$:

$$(b \wedge c) \vee d = a \vee d = d.$$
$$(b \vee d) \wedge (b \vee c) = e \vee e = e.$$

Таким чином $(b \wedge c) \vee d \neq (b \vee d) \wedge (b \vee c)$. Це означає, що дані гратки не ізоморфні ніякій підалгебрі алгебри підмножин, оскільки в алгебрі підмножин для будь-яких елементів B, C, D справедливі співвідношення:

$$(B \cap C) \cup D = (B \cup D) \cap (B \cup C),\ (B \cup C) \cap D = (B \cap D) \cup (B \cap C).$$

Групи і напівгрупи

Означення (Напівгрупа). Алгебра $<A; * >$ називається *напівгрупою*, якщо операція * задовольняє властивості асоціативності, тобто для будь-яких елементів $a, b, c \in A$ виконується $a * (b * c) = (a * b) * c$.

Означення (Моноїд). Алгебра $<A; *, 1 >$ називається *моноїдом*, якщо операція * задовольняє властивості асоціативності, а 1 є одиничним елементом щодо операції *, тобто для будь-якого елемента $a \in A$ виконується $1 * a = a * 1 = a$.

Приклад (Моноїд). Покажемо, що множина усіх послідовностей символів $a, b, c, \ldots$, включаючи порожню послідовність, з операцією *конкатенації* (приписування слів) і з порожньою послідовністю як одиничний елемент є моноїдом. Дійсно:

-дана множина замкнута щодо операції конкатенації;

-операція конкатенації очевидно асоціативна;

-порожнє слово є одиницею щодо операції конкатенації: приписування порожнього слова не змінює вихідного слова.

Множина усіх таких послідовностей називається множиною *слів в алфавіті* $\{a, b, c, \ldots\}$.

Теорема (Представлення скінченних моноїдів). Будь-який скінченний моноїд, що ізоморфний моноїду, носієм якого є множина відображень деякої скінченної множини на себе, операція * є композицією відображень, а 1 є *тотожним відображенням*.

Означення (Група). Алгебра $<A; *, ^{-1}, 1 >$ називається *групою*, якщо операція * задовольняє властивості асоціативності, 1 є одиничним елементом щодо операції *, a^{-1} є унарною операцією, що дає обернений елемент щодо операції *, тобто для будь-якого елемента $a \in A$ виконується співвідношення $a * a^{-1} = a^{-1} * a = 1$.

Теорема (Представлення скінченних груп). Будь-яка скінченна група ізоморфна деякій групі, носієм якої є множина, операцією * є добуток підстановок, 1 є тотожною підстановкою, операцією $^{-1}$ є обернена підстановка.

2 БУЛЕВІ ФУНКЦІЇ

2.1 Поняття булевої функції

В курсі математичного аналізу вивчаються функції, область визначення яких є деякими неперервними множинами. В курсі ж дискретної математики вивчаються функції, область визначення яких є *дискретними* множинами, найпростішою, однак нетривіальною з яких, є множина з двох елементів.

Означення (Булева функція). *Булевою функцією* від n аргументів називається функція f з *n-го* степеня множини $\{0, 1\}$ у множину $\{0, 1\}$. Інакше кажучи, булева функція – це функція, аргументи і значення якої належать множині $\{0, 1\}$. Множину $\{0, 1\}$ будемо надалі позначати як B.

Булеву функцію від n аргументів можна розглядати як *n-місну* алгебраїчну операцію на множині B. При цьому алгебра $<B; \Omega>$, де Ω – множина усіх можливих булевих функцій, називається *алгеброю логіки.*

Скінченність області визначення функції має важливу перевагу – такі функції можна задавати перерахуванням значень для різних значень аргументів. Тобто, щоб задати функцію від n змінних, треба визначити значення функції для кожного з 2^n її наборів. Ці значення зручно записувати у вигляді таблиці, у лівій частині якої розташовуються у лексико-графічному порядку двійкові вектори довжини n, а у правій частині – значення функції на них

X_1	X_2	...	x_{n-1}	x_n	F
0	0	...	0	0	f(0,0,...,0,0)
0	0	...	0	1	f(0,0,...,0,1)
0	0	...	1	0	f(0,0,...,1,0)
0	0	...	1	1	f(0,0,...,1,1)
...	...	...	...	...	...
1	1	...	0	0	f(1,1,...,0,0)
1	1	...	0	1	f(1,1,...,0,1)
1	1	...	1	0	f(1,1,...,1,0)
1	1	...	1	1	f(1,1,...,1,1)

Для будь-якого фіксованого впорядкування наборів логічна функція n змінних повністю визначається вектор-стовпцем своїх значень, розмірністю 2^n. Звідси число $|P_2(n)|$ різних функцій n змінних дорівнює числу різних двійкових векторів довжини 2^n, тобто 2^{2^n}. Набори, на яких $f = 1$, називають одиничними наборами функції f, а множина одиничних наборів - *одиничною множиною* f. Відповідно набори, де $f = 0$, називають *нульовими наборами.*

Множина B містить два елементи – їх можна розглядати як булеві функції від нуля, тобто порожньої множини, і таких змінних як *константа* нуля і *константа* одиниці.

Функцій від однієї змінної чотири: це константа нуля, константа одиниці, *тотожна функція*, тобто функція, значення якої збігається з аргументом і функція "*заперечення*". Заперечення будемо позначати символом $\bar{\ }$ як унарну операцію. Наведемо таблицю цих чотирьох функцій:

x	0	x	$\bar{x}$	1
0	0	0	1	1
1	0	1	0	1

Функції від деякого числа змінних можна розглядати як функції від більшого числа змінних. При цьому значення функції не змінюється під час зміни цих "додаткових" змінних. Такі змінні називаються *фіктивними*, на відміну від інших, що є *істотними*.

Означення (Фіктивні й істотні змінні). Змінна x_i називається *фіктивною* (несуттєвою) змінною функції $f(x_1,\cdots,x_n)$, якщо

$$f(x_1,\cdots,x_{i-1},0,x_{i+1},\cdots,x_n)=f(x_1,\cdots,x_{i-1},1,x_{i+1},\cdots,x_n)$$

для будь-яких значень $x_1,\cdots,x_{i-1},x_{i+1},\cdots,x_n$.
Інакше змінна x_i називається істотною.

Є шістнадцять функцій від двох аргументів. Найбільш вживані з них, тобто ті, що істотно залежать від обох змінних, наведені у таблиці нижче.

x_1	X_2	$x_1 \& x_2$	$x_1 \vee x_2$	$x_1 \supset x_2$	$x_1 \oplus x_2$	$x_1 \equiv x_2$	$x_1 \mid x_2$
0	0	0	0	1	0	1	0
0	1	0	1	1	1	0	1
1	0	0	1	0	1	0	1
1	1	1	1	1	0	1	1

Ці функції записуються як бінарні операції. Операція $x_1 \& x_2$ називається *кон'юнкцією*, операція $x_1 \vee x_2$ – *диз'юнкцією*, операція $x_1 \supset x_2$ (зустрічається ще як $x_1 \to x_2$) – *імплікацією*, операція $x_1 \equiv x_2$ – *еквівалентністю*, операція $x_1 \oplus x_2$ – *сумою за модулем два*, операція $x_1 \mid x_2$ – *функцією штрих-Шефера*.

Значення нуля і одиниці часто інтерпретують як "хибність" і "істину". Звідси зрозумілим стає назва функції "заперечення" – вона змінює "хибність" на "істину", а "істину" на "хибність". Заперечення читається як "ні". Кон'юнкція читається як "і". Дійсно, кон'юнкція дорівнює одиниці тоді і тільки тоді, коли рівні одиниці і перша і друга змінні. Крім $x_1 \& x_2$ часто використовують позначення $x_1 \wedge x_2$ або $x_1 \cdot x_2$ або x_1x_2 або $\min(x_1, x_2)$.

Диз'юнкція читається як "або", тобто диз'юнкція дорівнює одиниці тоді і тільки тоді, коли рівні одиниці перша або друга змінні. Імплікація описує той факт, що з x_1 випливає x_2.

2.2 Суперпозиції функцій і формули

Означення (Суперпозиція функцій і формули)

Суперпозицією функцій $f_1,\ldots,f_m$ називається функція f, яка отримана підстановкою цих функцій одна в іншу й перейменування змінних.

Формулою називається вираз, що описує суперпозицію функцій.

Приклади (Суперпозиції і формули функцій):

- функція $f(x,y) = \overline{(x \,\&\, y)}$ є суперпозицією функцій $^{-}$ і $\&$;
- функція $g(x,y) = x \oplus (x \vee y)$ є суперпозицією функцій $\oplus$ і $\vee$;
- функція $h(x, y, z) = (x \,\&\, y) \oplus z$ є суперпозицією функцій $\oplus$ і $\&$.

Нехай задана множина (скінченна або нескінченна) вихідних функцій $\Sigma=\{f_1,\ldots,f_m,\ldots\}$. Символи змінних $x_1,\ldots, x_n,\ldots$ будемо вважати *формулами глибини нуль*. Формула F має глибину $k+1$, якщо F має вигляд $f_i(F_1,\ldots,F_{n_i})$, де $f_i \in \Sigma$, n_i - число аргументів f_i, а $F_1,\ldots, F_{n_i}$ - формули, максимальна з глибин яких дорівнює k. $F_1, \ldots, F_{n_i}$ - називаються *підформулами* F; f_i - називається *зовнішньою або головною операцією* формули F. Наприклад, $f_2(x_1, x_2, x_3)$ - це формула глибини один. $f_3(f_1(x_3, x_1), f_2(x_1, f_3(x_1,x_2)))$ - формули глибини три. Надалі формули будемо записувати в їх звичному вигляді, де знаки функцій стоять між аргументами. Такий запис називають інфіксним. Так, наприклад, якщо f_1 - диз'юнкція; f_2 - кон'юнкція; f_3 - додавання за модулем два, то наведена вище формула буде мати такий вигляд

$$(x_3 \vee x_1) \oplus (x_1 \,\&\, (x_1 \oplus x_2)).$$

Всі формули, що побудовані таким чином, і містять символи змінних, дужки і знаки функцій множини Σ, називаються *формулами над* Σ.

Про формулу, що описує функцію, говорять, що вона реалізує цю функцію. На відміну від табличного завдання опис функцій формулою не єдиний. Наприклад, функцію Шеффера можна описати формулою:

$$\overline{x_1} \vee \overline{x_2} \text{ або } \overline{x_1 x_2},$$

а функцію стрілки Пірса – формулами: $\overline{x_1}\,\overline{x_2}$ або $\overline{x_1 \vee x_2}$.

Формули, що описують одну і ту ж саму функцію, називають *еквівалентними* або *рівносильними*. Наприклад, $\psi_{14}(x_1,x_2) = \overline{x_1} \vee \overline{x_2} = \overline{x_1 x_2}$.

Наведені нижче співвідношення можуть бути перевірені прямим порівнянням значень функцій у лівій і правій частинах співвідношень на усіх можливих наборах аргументів.

$$x \vee y = y \vee x; \qquad x \oplus y = y \oplus x;$$
$$x \,\&\, (y \,\&\, z) = (x \,\&\, y) \,\&\, z; \qquad x \vee (y \vee z) = (x \vee y) \vee z;$$
$$x \oplus (y \oplus z) = (x \oplus y) \oplus z; \qquad x \vee (y \,\&\, z) = (x \vee y) \,\&\, (x \vee z);$$
$$x \,\&\, (y \vee z) = (x \,\&\, y) \vee (x \,\&\, z); \qquad x \,\&\, x = x;$$
$$\overline{(x \,\&\, y)} = \bar{x} \vee \bar{y}; \qquad \overline{(x \vee y)} = \bar{x} \,\&\, \bar{y};$$
$$x \equiv y = (x \,\&\, y) \vee (\bar{x} \,\&\, \bar{y}); \qquad x \,\&\, \bar{x} = 0;$$
$$x \,\&\, 0 = 0; \qquad x \,\&\, 1 = x;$$
$$x \vee x = x; \qquad x \vee \bar{x} = 1;$$
$$x \vee 0 = x; \qquad x \vee 1 = 1;$$
$$x \oplus y = (x \,\&\, \bar{y}) \vee (\bar{x} \,\&\, y); \qquad x \supset y = \bar{x} \vee y.$$

2.3 Двоїсті функції

Симетрія 0 і 1 на множині B приводить до поняття двоїстості.

Означення (Двоїста функція). Функція $g(x_1,\dots,x_n) = \overline{f(\overline{x_1},\dots,\overline{x_n})}$ називається *двоїстою функцією* до функції f і позначається f^*.

Приклад (Двоїсті функції). $(x \,\&\, y)^* = \overline{(\bar{x} \,\&\, \bar{y})} = x \vee y$.

Означення (Двоїста до двоїстої функції). Функція, що двоїста до двоїстої функції f дорівнює самій функції f.

Доведення. $f^*(x_1,\dots,x_n)^* = \overline{f(\overline{x_1},\dots,\overline{x_2})}^* = \overline{\overline{f(\overline{\overline{x_1}},\dots,\overline{\overline{x_n}})}} = f(x_1,\dots,x_n)$.

Розглянемо, що відбувається з таблицею двоїстої функції. Заміна набору $(x_1,\dots,x_n)$ на $(\bar{x}_1,\dots,\bar{x}_n)$ відповідає "перевертанню" таблиці. Дійсно, набори $(x_1,\dots,x_n)$ і $(\bar{x}_1,\dots,\bar{x}_n)$ розташовані симетрично щодо середини таблиці. Тепер залишається застосувати операцію заперечення до результату функції, тобто поміняти 0 на 1 і 1 на 0. Таким чином вектор значень функції, двоїстої до вихідної, отримується з вектора вихідної функції перевертанням і заміною 0 на 1, а 1 на 0.

Приклад (Вектор двоїстої функції). Функції $x \,\&\, y$ і $x \vee y$, що задаються векторами значень (0,0,0,1) і (0,1,1,1), двоїсті одна до одної. Також двоїстими є $x \oplus y$ і $x \equiv y$, що задаються векторами (0,1,1,0) і (1,0,0,1). Кожна з функцій x і $\bar{x}$ (вектори (0,1) і (1,0) відповідно) двоїста сама собі.

Теорема (Принцип двоїстості). Функція, що двоїста до суперпозиції функцій, дорівнює суперпозиції двоїстих функцій. Точніше:

$$f_0(f_1,\dots,f_m)^* = f_0^*(f_1^*,\dots,f_m^*).$$

Доведення. $$(f_1(x_1,\dots,x_n),\dots,f_m(x_1,\dots,x_n))^* =$$

$$= \overline{f_0(f_1(\overline{x_1},\dots,\overline{x_n}),\dots,f_m(\overline{x_1},\dots,\overline{x_n}))} =$$

$$= \overline{f_0(\overline{\overline{f_1(\overline{x_1},\dots,\overline{x_n})}},\dots,\overline{\overline{f_m(\overline{x_1},\dots,\overline{x_n})}})} =$$

$$= \overline{f_0(\overline{f_1{}^*(x_1,\dots,x_n)},\dots,\overline{f_m{}^*(x_1,\dots,x_n)})} =$$

$$= f_0^*(f_1^*(x_1,\dots,x_n),\dots,f_m^*(x_1,\dots,x_n))$$

2.4 Розкладання функцій за змінними

Введемо позначення:

$$x^\sigma = \begin{cases} \overline{x}, \text{ якщо } \sigma = 0 \\ x, \text{ якщо } \sigma = 1 \end{cases}$$

Теорема (Розкладання у диз'юнкцію). Будь-яку функцію $f(x_1,\dots,x_m)$ для будь-якого n ($1 \le n \le m$) можна описати у вигляді

$$f(x_1,\dots,x_m) = \bigvee_{(\sigma_1,\dots,\sigma_n)} x_1^{\sigma_1} \mathbin{\&} \dots \mathbin{\&} x_n^{\sigma_n} \mathbin{\&} f(\sigma_1,\dots,\sigma_n, x_{n+1},\dots,x_m).$$

Доведення. Покажемо, що для будь-якого набору значень змінних $(x_1,\dots,x_n,x_{n+1},\dots,x_m)$ значення лівої і правої частин збігаються. Візьмемо фіксований набір $(x_1,\dots,x_n,x_{n+1},\dots,x_m)$. Розглянемо вираз $x_1^{\sigma_1} \mathbin{\&} \dots \mathbin{\&} x_n^{\sigma_n}$. Якщо одне зі значень $x_i^{\sigma_i}$ дорівнює нулю, то і весь вираз дорівнює нулю. Тоді і вираз $x_1^{\sigma_1} \mathbin{\&} \dots \mathbin{\&} x_n^{\sigma_n} \mathbin{\&} f(\sigma_1,\dots,\sigma_n,x_{n+1},\dots,x_m)$ дорівнює нулю. Одиниці ж вираз $x_1^{\sigma_1} \mathbin{\&} \dots \mathbin{\&} x_n^{\sigma_n}$ дорівнює тільки в тому випадку, якщо $\sigma_1 = x_1$, ..., $\sigma_n = x_n$. При цьому $f(\sigma_1,\dots,\sigma_n,x_{n+1},\dots,x_m) = f(x_1,\dots,x_n,x_{n+1},\dots,x_m)$. Таким чином, значення правої частини завжди дорівнює $f(x_1,\dots,x_m)$, тобто значенню лівої частини.

Теорема. (Розкладання у кон'юнкцію). Будь-яку функцію $f(x_1,\dots,x_m)$ для будь-якого n ($1 \le n \le m$) можна описати у вигляді

$$f(x_1,\dots,x_m) = \mathop{\&}_{(\sigma_1,\dots,\sigma_n)} x_1^{\neg\sigma_1} \vee \dots \vee x_n^{\neg\sigma_n} \vee f(\sigma_1,\dots,\sigma_n,x_{n+1},\dots,x_m).$$

Розкладання по всіх змінних дають суперпозицію кон'юнкції, диз'юнкції і заперечення.

Наслідок 1. (Досконала диз'юнктивна нормальна форма). Будь-яка функція f може бути описана у такій формі:

$$f(x_1,...,x_m) = \bigvee_{(\sigma_1,...,\sigma_m)} x_1^{\sigma_1} \& ... \& x_m^{\sigma_m} \& f(\sigma_1,...,\sigma_m) =$$

$$= \bigvee_{(\sigma_1,...,\sigma_m):f(\sigma_1,...,\sigma_m)=1} x_1^{\sigma_1} \& ... \& x_m^{\sigma_m}.$$

Наслідок 2. (Досконала кон'юнктивна нормальна форма). Будь-яка функція f може бути описана у такій формі:

$$f(x_1,...,x_m) = \mathop{\&}_{(\sigma_1,...,\sigma_m):f(\sigma_1,...,\sigma_m)=0} x_1^{\bar{\sigma}_1} \vee ... \vee x_m^{\bar{\sigma}_m}.$$

Таким чином, будь-яка булева функція може бути описана суперпозицією кон'юнкції, диз'юнкції і заперечення. Розкладання за всіма змінними в диз'юнкцію називається *досконалою диз'юнктивною нормальною формою* (ДДНФ) функції, а розкладання в кон'юнкцію – *досконалою кон'юнктивною нормальною формою* (ДКНФ).

Досконала диз'юнктивна і кон'юнктивна нормальна форми дають спосіб описування булевої функції через суперпозицію кон'юнкції, диз'юнкції і заперечення якщо в нас є таблиця значень функції.

Щоб одержати досконалу диз'юнктивну нормальну форму, треба взяти всі набори, на яких значення функції дорівнює одиниці і записати для кожного з них кон'юнкцію змінних і їх заперечень. Якщо в наборі значення деякої змінної дорівнює нулю, то цю змінну треба взяти із запереченням, а якщо одиниці, то її треба взяти без заперечення. З отриманих кон'юнкцій треба побудувати диз'юнкцію.

Щоб одержати ДКНФ, треба узяти всі набори, на яких значення функції дорівнює нулю і записати для кожного з них диз'юнкцію змінних і їх заперечень. Якщо в наборі значення деякої змінної дорівнює нулю, то цю змінну треба взяти без заперечення, якщо одиниці, то її треба взяти із запереченням. З диз'юнкцій, що вийшли, треба побудувати кон'юнкцію.

Приклад. Побудуємо досконалу диз'юнктивну нормальну форму функції задану у наведеній нижче таблиці.

x	y	z	f	x	y	z	f
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	1	1	1	1	1	1

Набори, на яких функція дорівнює одиниці – це (0,1,1), (1,0,1), (1,1,0), (1,1,1). Перший набір дає кон'юнкцію $\bar{x}$ & y & z, другий – x & $\bar{x}$& & z, третій – x & y & $\bar{z}$, четвертий – x & y & z. У результаті одержуємо $(\bar{x} \& y \& z) \vee (x \& \bar{y} \& z) \vee (x \& y \& \bar{z}) \vee (x \& y \& z)$.

2.5 Еквівалентні перетворення булевих функцій

Нехай функція f_1 задана формулою $F1$, а функція f_2 - формулою F_2. Підстановка F_1 і F_2 у диз'юнкцію $x_1 \vee x_2$ дає формулу $F_1 \vee F_2$. Нехай F_1' і F_2' еквівалентні відповідно F_1 і F_2. Тоді отримаємо формулу $F_1' \vee F_2'$ еквівалентну $F_1 \vee F_2$. Це означає, що диз'юнкцію можна розглядати як бінарну операцію на множині логічних функцій, яка кожній парі функцій f_1 і f_2 незалежно від виду формул, у якій вони описані, однозначно ставить у відповідність функцію $f_1 \vee f_2$. Аналогічно і кон'юнкцію можна розглядати як бінарну операцію, а заперечення - як унарну операцію над функціями.

Алгебра $(P_2; \vee, \&, \bar{\ })$, основною множиною якої є уся множина логічних функцій, називається *булевою алгеброю логічних функцій.*

Основні властивості булевих операцій:

- **Асоціативність:**
  $x_1 (x_2 x_3) = (x_1 x_2) x_3$; $(x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3)$.
- **Комутативність:** $x_1 x_2 = x_2 x_1$; $x_1 \vee x_2 = x_2 \vee x_1$.
- **Дистрибутивність кон'юнкції щодо диз'юнкції:**
  $x_1 (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3$.
- **Дистрибутивність диз'юнкції щодо кон'юнкції:**
  $x_1 \vee (x_2 x_3) = (x_1 \vee x_2)(x_1 \vee x_3)$.
- **Ідемпотентність:** $x x = x$; $x \vee x = x$.
- **Подвійне заперечення:** $\bar{\bar{x}} = x$.
- **Властивості констант:**
  $x \& 1 = x$; $x \& 0 = 0$; $x \vee 1 = 1$; $x \vee 0 = x$; $\bar{0} = 1$; $\bar{1} = 0$.
- **Правила де Моргана:** $\overline{x_1 x_2} = \bar{x}_1 \vee \bar{x}_2$; $\overline{x_1 \vee x_2} = \bar{x}_1 \bar{x}_2$.
- **Закон протиріччя:** $x\bar{x} = 0$.
- **Закон "виключеного третього":** $x \vee \bar{x} = 1$.

Еквівалентні формули дозволяють здійснити заміну однієї формули еквівалентною їй іншою формулою. Це твердження є *правилом заміни підформул*. Відзначимо різницю між правилами підстановки і заміни. Під час підстановки змінні заміняються на формулу. Формула може бути будь-якою, але необхідна одночасна її підстановка замість усіх входжень змінної. Під час заміни підформул може бути замінена будь-яка підформула, проте не на будь-яку іншу, а тільки на еквівалентну їй. При цьому заміна усіх входжень вихідної підформули не обов'язкова.

Приклад. Нехай маємо $\overline{x_1x_2}=\overline{x_1}\vee\overline{x_2}$. Замість x_1 підставимо $\overline{x_1}x_3$. Отримаємо таку формулу : $\overline{\overline{x_1}x_3x_2}=\overline{\overline{x_1}x_3}\vee\overline{x_2}$, тобто дві формули не еквівалентні вихідній, але еквівалентні між собою. Потім в правій частині отриманого співвідношення $\overline{\overline{x_1}x_3}$ замінимо еквівалентною формулою $\overline{\overline{x_1}}\vee\overline{x_3}$, а в отриманому виразі $\overline{\overline{x_1}}$ замінимо на x. Тоді усі формули в отриманій послідовності перетворень:

$$\overline{\overline{x_1}x_3x_2}\Rightarrow\overline{\overline{x_1}x_3}\vee\overline{x_2}\Rightarrow\overline{\overline{x_1}}\vee\overline{x_3}\vee\overline{x_2}\Rightarrow x_1\vee\overline{x_3}\vee\overline{x_2}$$

будуть еквівалентні. Такі перетворення називаються *еквівалентними перетвореннями.*

Розглянемо основні еквівалентні перетворення булевої алгебри. Відзначимо, що у булевій алгебрі прийнято опускати дужки аналогічно їх опусканню для операції множення в арифметичних формулах, тобто при послідовному виконанні декількох кон'юнкцій або диз'юнкцій і коли вони є зовнішніми дужками в кон'юнкції.

- **Спрощення формул:**

а) ***Поглинання:***

$$x\vee xy=x;\ x(x\vee y)=x.$$

Для доведення першої рівності застосовуються послідовно перша властивість констант, дистрибутивність кон'юнкції щодо диз'юнкції, третя і потім перша властивість констант:

$$x\vee xy=x\,\&\,1\vee xy=x(1\vee e)=x\,\&\,1=x.$$

Друга рівність доводиться послідовним застосуванням дистрибутивності кон'юнкції щодо диз'юнкції, ідемпотентності і першої рівності:

$$x(x\vee y)=xx\vee xy=x\vee xy=x.$$

b) ***Склеювання:*** $xy\vee x\overline{y}=x$.

Доведення: $xy\vee x\overline{y}=x(y\vee\overline{y})=x\,\&\,1=x$.

c) ***Узагальнене склеювання:*** $xz\vee y\overline{z}\vee xy=xz\vee y\overline{z}$.

Доводиться за допомогою розщеплення, тобто застосовуючи склеювання в зворотну сторону, і другої рівності поглинання:

$$xz\vee y\overline{z}\vee xy=xz\vee y\overline{z}\vee xyz\vee xy\overline{z}=xz\vee y\overline{z}$$

г) $x \vee \bar{x}y = x \vee y$.

• Приведення до диз'юнктивної нормальної форми (ДНФ)

Означення. *Елементарними кон'юнкціями* називаються кон'юнкції змінних або їх заперечень, у яких кожна змінна зустрічається не більше одного разу. ДНФ називається формула, що має вигляд диз'юнкції елементарних кон'юнкцій.

Приведення до ДНФ здійснюється таким чином. Спочатку у відповідності з властивістю подвійного заперечення і правил де Моргана усі заперечення "опускаються" до змінних. Потім розкриваються дужки, у відповідності з властивістю ідемпотентності, закону протиріччя і закону "виключеного третього" видаляються зайві кон'юнкції і повторення змінних у кон'юнкціях. Далі знищуються константи.

Приклад. $xy \vee \bar{x}(y \vee xz)\left(\overline{x(\bar{y} \vee z) \vee yz}\right) = xy \vee (\bar{x}y \vee \bar{x}xz)\left(\bar{x} \vee \overline{(\bar{y} \vee z)}\right)yz =$
$= xy \vee \bar{x}y(\bar{x} \vee y\bar{z})(\bar{y} \vee \bar{z}) = xy \vee \bar{x}y(\bar{x}\bar{y} \vee y\bar{y}\bar{z} \vee \bar{x}\bar{z} \vee y\bar{z}) = xy \vee \bar{x}y\bar{z} =$
$= y(x \vee \bar{x}\bar{z}) = xy \vee y\bar{z}$.

Будь-яку ДНФ можна привести до ДДНФ розщепленням кон'юнкцій, які містять не всі змінні. Наприклад:

$$xy \vee \bar{x}y\bar{z} = xyz \vee xy\bar{z} \vee \bar{x}y\bar{z}.$$

Якщо з формули F_1 за допомогою еквівалентних перетворень можна отримати формулу F_2, то із F_2 застосовуючи ті ж самі еквівалентні співвідношення можна отримати F_1, тобто будь-яке еквівалентне перетворення є оберненим.

Теорема. Для будь-яких двох еквівалентних формул F_1 і F_2 існують еквівалентні перетворення F_1 у F_2, які основані на співвідношеннях основних властивостей булевих операцій. Дійсно, перетворимо F_1 і F_2 у ДДНФ. Оскільки F_1 і F_2 еквівалентні, то їх ДДНФ збігаються. Обертаючи друге перетворення, отримаємо перетворення:

$$F_1 \Rightarrow \text{ДДНФ} \Rightarrow F_2.$$

Приведення до кон'юнктивної нормальної форми (КНФ)

Аналогічно ДНФ КНФ визначається як кон'юнкція елементарних диз'юнкцій. Від ДНФ до КНФ перейдемо таким чином. Нехай ДНФ F має вигляд $F = к_1 \vee \ldots \vee к_m$, де $к_1, \ldots, к_m$ - елементарні кон'юнкції. Формулу $\overline{k_1 \vee \ldots \vee k_m}$ приведемо до ДНФ $к'_1 \vee \ldots \vee к'_m$. Тоді

$$F = \overline{\overline{k_1 \vee \ldots \vee k_m}} = \overline{k'_1 \vee \ldots \vee k'_m} = k'_1 k'_2 \ldots k'_m .$$

За правилом де Моргана заперечення елементарних кон'юнкцій перетворюються в елементарні диз'юнкції, що і дає КНФ.

Приклад. $x\bar{y} \vee \bar{x}y \vee x\bar{z} = \overline{\overline{x\bar{y} \vee \bar{x}y \vee x\bar{z}}} = \overline{(\bar{x} \vee y)(x \vee \bar{y})(\bar{x} \vee z)} =$

$$= \overline{\bar{x}\bar{y} \vee xyz} = (x \vee y)(\bar{x} \vee \bar{y} \vee z).$$

Аналогом ДДНФ є досконала кон'юнктивна нормальна форма (ДКНФ), кожна елементарна функція якої містить усі змінні. Єдина функція, що не має ДКНФ, - константа одиниці.

2.6 Булева алгебра і теорія множин

Будь-яка алгебра типу (2, 2, 1) називається булевою алгеброю, якщо її операції задовольняють співвідношення властивостей булевих операцій. В алгебрі множин елементами є підмножини фіксованої множини U, операція & відповідає операції перетину, $\vee$ - об'єднання, операція $\bar{\ }$ відповідає доповненню, 1 є саме множина U, а 0 - порожня множина.

Пропонується показати самостійно справедливість співвідношень, які описують властивості булевих операцій для алгебри множин. Для цього змінні потрібно розглядати як множини.

Теорема. Якщо $|U| = n$, то булева алгебра $(\Re(u); \cap, \cup, \bar{\ })$ ізоморфна булевій алгебрі $(B_n; \vee, \&, \bar{\ })$.

Теорема. Якщо $|U| = 2^m$, то булева алгебра $(\Re(u); \cap, \cup, \bar{\ })$ ізоморфна булевій алгебрі функцій $(P_2(m); \vee, \&, \bar{\ })$.

Відзначимо ще один факт, який зв'язує логічні функції з основними поняттями теорії множин: якщо $(f \to g) \equiv 1$, то $M_f \subseteq M_g$. Дійсно, якщо $(f \to g) \equiv 1$, то за визначенням імплікації (функція φ_{13}) випливає, що ні для якого набору σ не може бути одночасно $f(\sigma) = 1$ і $g(\sigma) = 0$. Тому, якщо $f(\sigma) = 1$, то $g(\sigma) = 1$, тобто, якщо $\sigma \in M\mathrm{f} \to \sigma \in M\mathrm{g}$, то $M_f \subseteq M_g$. У такому випадку говорять, що функція f *імплікує* функцію g.

Якщо f елементарна кон'юнкція, то f називається *імплікантом* g. А якщо після видалення змінної f перестає бути імплікантом g, то f називається *простим імплікантом* g.

Наприклад, для функції $x(y \vee z)$ кон'юнкції xy і xz є прості імпліканти, а xyz – імплікант, але не простий. Відзначимо, що будь-яка кон'юнкція будь-якої ДНФ даної функції є імплікантом цієї функції.

2.7 Повнота і замкненість

Дотепер розглядались два способи задання логічних функцій - табличний і формулою. Табличний спосіб, який задає відповідність між двійковими наборами і значеннями функції на цих наборах є універсальним, проте громіздким. Формула є компактним записом, але вона задає функцію через інші функції. Тому для будь-якої системи функцій Σ виникає природне запитання: чи будь-яка логічна функція може бути описана формулою у системі Σ? Више вказувалось, що система $\Sigma_0 = \{\&, \vee, \bar{\ }\}$ задовольняє ці вимоги. Покажемо, як вирішується це питання для довільної системи Σ.

Функціонально повні системи. Система функцій Σ називається *функціонально повною системою,* якщо будь-яка логічна функція може бути описана формулою у системі Σ, тобто є суперпозицією функцій із системи Σ. Наприклад:

1. Система $\Sigma_0 = \{\&, \vee, \bar{\ }\}$ функціонально повна.

2. Системи $\Sigma_1 = \{\&, \bar{\ }\}$ і $\Sigma_2 = \{\vee, \bar{\ }\}$ також функціонально повні. Дійсно, із законів де Моргана і подвійного заперечення випливає, що у кожній з них відсутня щодо системи Σ_0 функція може бути описана через дві інші:

$$x_1 \vee x_2 = \overline{\overline{x_1} \,\&\, \overline{x_2}}\,,\quad x_1 \,\&\, x_2 = \overline{\overline{x_1} \vee \overline{x_2}}$$

Булева формула $x_1 x_2 \vee \overline{x_2}(x_3 \vee x_4)$ у системі Σ_1 перейде у формулу $\overline{\overline{x_1 x_2}\,\overline{\overline{x_2}\,\overline{x_3}\,\overline{x_4}}}$, а у системі Σ_2 – у формулу $\overline{\overline{x_1} \vee \overline{x_2}} \vee \overline{x_2 \vee \overline{x_3 \vee x_4}}$.

3. Системи $\Sigma_3 = \{|\}$ (штрих Шеффера) і $\Sigma_4 = \{\downarrow\}$ (стрілка Пірса) функціонально повні, оскільки із $\overline{x_1} \vee \overline{x_2}$ або $\overline{x_1 x_2}$ і $\overline{x_1}\,\overline{x_2}$ або $\overline{x_1 \vee x_2}$ випливає, що:

$$\bar{x} = x \mid x = x \downarrow x;\; x_1 \vee x_2 = \overline{x_1 \downarrow x_2} = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2);$$
$$x_1 x_2 = \overline{x_1 | x_2} = (x_1 | x_2)|(x_1 | x_2).$$

Таким чином Σ_3 зводиться до Σ_1, а Σ_4 - до Σ_2

3. Система $\Sigma_5 = \{\&, \oplus, 1\}$ функціонально повна, оскільки $\bar{x} = x \oplus 1$ і тому Σ_5 зводиться до Σ_1. Розглянемо детальніше її властивості.

Алгебра Жегалкіна і лінійні функції. Алгебра над множиною логічних функцій з двома бінарними операціями & і ⊕ називається *алгеброю Жегалкіна.* В алгебрі Жегалкіна виконуються співвідношення:

$$x \oplus y = y \oplus x;\; x(y \oplus z) = xy \oplus xz;\; x \oplus x = 0;\; x \oplus 0 = x$$

а також співвідношення булевої алгебри щодо кон'юнкції ідемпотентності і констант: $x_1(x_2x_3) = (x_1x_2)x_3$; $x_1x_2 = x_2x_1$; $xx = x$; $x\,\&1 = x$; $x\&\,0 = 0$.

Заперечення і диз'юнкція виражаються таким чином:

$$\bar{x} = x \oplus 1;\ x \vee y = \overline{\bar{x}\,\bar{y}} = (x \oplus 1)(y \oplus 1) \oplus 1 = xy \oplus x \oplus y.$$

Якщо в будь-якій формулі алгебри Жегалкіна розкрити дужки і зробити всі можливі спрощення, то отримаємо формулу, що має вигляд суми добутків, тобто поліном за *mod* 2. Така формула називається *поліномом Жегалкіна* для даної функції.

Приклади.

1. $x_1\overline{x_2} \vee \overline{x_1}x_3 = x_1\overline{x_2} \oplus \overline{x_1}x_3 = x_1(x_2 \oplus 1) \oplus (x_2 \oplus 1)x_3 =$
$= x_1x_2 \oplus x_1x_3 \oplus x_1 \oplus x_3$.
2. $x_1x_2 \vee \overline{x_1}\,\overline{x_2} = x_1x_2 \oplus (x_1 \oplus 1)(x_2 \oplus 1) = x_1 \oplus x_2 \oplus 1$.

Теорема. Для будь-якої логічної функції існує поліном Жегалкіна, причому єдиний.

Доведення. Існування полінома уже доведено вище. Для доведення його єдиності достатньо показати взаємно-однозначну відповідність між множиною усіх функцій від n змінних і множиною всіх поліномів Жегалкіна від n змінних. Число різних членів (тобто кон'юнкцій змінних) поліномів від n змінних – 2^n (порожній підмножині відповідає член 1), і число різних поліномів, що можна утворити з цих кон'юнкцій, дорівнює числу всіх підмножин множини кон'юнкцій, тобто 2^n (порожній підмножині кон'юнкцій відповідає поліном 0).

Таким чином, число всіх поліномів Жегалкіна від n змінних дорівнює числу всіх функцій від n змінних. Оскільки різним функціям відповідають різни поліноми, то це означає, що множинам функцій від n змінних і поліномом встановлена взаємно однозначна відповідність.

Функція, для якої поліном Жегалкіна має вигляд $\sum \alpha_i \oplus \gamma$, де α_i, γ рівні 0 і 1, називається *лінійною.* Всі функції від однієї змінної лінійні. Лінійними функціями від двох змінних є сума за *mod* 2 і еквівалентність.

Замкнені класи. Монотонні функції. Множина M логічних функцій називається *замкненим класом,* якщо будь-яка суперпозиція функцій із M знову належить M.

Будь-яка система Σ логічних функцій породжує деякий замкнений клас, тобто такий клас, який містить всі функції, які можна отримати суперпозиціями із Σ. Такий клас називається *замиканням* Σ і позначається як $[\Sigma]$. Очевидно, що якщо M - замкнений клас, то $[M]=M$, а якщо M - функціонально повна система, то $[M]=P_2$.

Приклади.

1. Множина всіх диз'юнкцій, тобто функцій виду $x_1 \vee, \ldots, \vee x_n$, є замкненим класом.

2. Множина всіх лінійних функцій є замкненим класом, оскільки підстановка формули $\sum \alpha_i \oplus \gamma$ у формулу такого ж виду знову дає формулу того самого виду.

Функція $f(x_1, \ldots, x_n)$ називається *монотонною,* якщо для будь-яких двійкових векторів σ і τ довжини n із того, що $\sigma \leq \tau \Rightarrow f(\sigma) \leq f(\tau)$. Клас монотонних функцій є замкненим.

Приклади.

1. Константи нуля, одиниці і функція x є монотонними, а функція заперечення $\bar{x}$ - немонотонна.
2. Диз'юнкція і кон'юнкція будь-якого числа змінних є монотонними функціями.
3. Розглянемо дві функції від трьох змінних, що задані у табл. нижче.

Функція f_1 немонотонна, оскільки 001<101, а $f_1(001) > f_1(101)$. Функція f_2 – монотонна (показати самостійно).

x_1	x_2	x_3	f_1	f_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	1
0	1	1	1	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

У загальному випадку аналіз булевих функцій у вигляді таблиць може бути надто громіздким. Тому для визначення монотонності функцій корисними є такі теореми.

Теорема. Будь-яка булева формула, що не містить заперечення, описує монотонну функцію, відмінну від нуля і одиниці, і навпаки, для будь-якої монотонної функції, відмінної від нуля і одиниці, знайдеться булева формула, що опису їх, без заперечення.

Теорема. Множина всіх монотонних функцій є замкненим класом.

Дана теорема випливає безпосередньо з попередньої і тієї очевидної підстави, що підстановка формул без заперечення у формули без заперечень знову дають формули без заперечень.

Наслідок. Клас монотонних функцій є замиканням системи функцій $\{\&,\vee,0,1\}$. Дане твердження випливає з того, що будь-яка булева формула без заперечень є суперпозицією диз'юнкцій і кон'юнкцій.

Теореми про функціональну повноту. Наведені вище теореми дають необхідні і достатні умови функціональної повноти для довільної системи функцій Σ.

Лема 1. **(про немонотонні функції).** Якщо функція $f(x_1,...,x_n)$ немонотонна, то підстановкою констант з неї можна одержати заперечення. Тобто існує така підстановка n-1 констант, коли функція одної змінної, що залишилася, є запереченням.

Лема 2. **(Про нелінійні функції).** Якщо функція $f(x_1,...,x_n)$ нелінійна, то за допомогою підстановки констант і використання заперечень з неї можна одержати диз'юнкцію і кон'юнкцію. Точніше, існує подання диз'юнкції і кон'юнкції як суперпозиції констант, заперечень і функції f.

Наведені вище дві леми дозволяють отримати всі булеві операції за допомогою немонотонних функцій, нелінійних функцій і констант. Однак це ще не функціональна повнота в звичайному змісті, оскільки константи з самого початку передбачаються даними.

Такі умови часто бувають виправданими, зокрема при синтезі логічних схем, де системі логічних функцій відповідає набір типових логічних елементів, а повнота системи означає можливість реалізації за допомогою елементів даної системи будь-яких логічних функцій.

Під час схемної реалізації константи нуля і одиниці спеціальних елементів не потребують. Тому є сенс ввести послаблене поняття функціональної повноти. Система функцій Σ називається функціонально повною у *слабкому змісті*, якщо будь яка логічна функція може бути подана формулою над системою $\Sigma\cup\{0,1\}$, тобто суперпозицією констант і функцій із Σ.

Очевидно, що із звичайної повноти системи випливає її слабка повнота.

Теорема. (Перша основна теорема про функціональну повноту). Для того, щоб система функцій Σ була функціонально повною в слабкому змісті, необхідно і достатньо, щоб вона містила хоча б одну немонотонну і хоча б одну нелінійну функцію.

Необхідність. Класи монотонних і лінійних функцій замкнені і містять 0 і 1. Тому, якщо Σ не містить немонотонних або нелінійних функцій, то їх не можна одержати як суперпозиції функцій із Σ і констант.

Достатність. Нехай Σ містить немонотонну і нелінійну функцію. Тоді за лемою 1 підстановкою констант з монотонної функції одержуємо заперечення, а потім за лемою 2 з нелінійної функції за допомогою заперечень і констант одержуємо диз'юнкцію і кон'юнкцію.

Приклади.

1. Система $\Sigma_6 = \{\&, \oplus\}$ функціонально повна в слабкому змісті, тому що кон'юнкція нелінійна, а сума по *mod* 2 немонотонна. Константа нуля отримується із співвідношення $x \oplus x = 0$, однак константу одиниці за допомогою кон'юнкції і суми за *mod* 2 одержати не можна, тому Σ_6 не є функціонально повною системою в звичайному (сильному) змісті.

2. У функціонально повній системі Σ_3 єдина функція штрих-Шеффера одночасно нелінійна і немонотонна.

3. Розглянемо систему Σ_7, яка складається з однієї функції f_1, заданої розглянутою вище таблицею. Немонотонність f_1 вже встановлена. Одержимо для неї поліномом Жегалкіна

$$f_1 = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3 = (x_1 \oplus 1)(x_2 \oplus 1) x_3 \oplus$$
$$\oplus (x_1 \oplus 1) x_2 x_3 \oplus x_1 (x_2 \oplus 1)(x_3 \oplus 1) \oplus$$
$$\oplus x_1 x_2 x_3 = x_1 x_2 \oplus x_1 \oplus x_3.$$

Тобто f_1 - нелінійна і система Σ_7 функціонально повна в слабкому змісті.

Для формулювання необхідних і достатніх умов сильної повноти розглянемо три замкнених класи.

Функція $f(x_1,\ldots,x_n)$ називається *функцією, що зберігає нуль*, якщо $f(0, 0, \ldots, 0) = 0$. Функція $f(x_1,\ldots,x_n)$ називається *функцією, що зберігає одиницю*, якщо $f(1, 1, \ldots, 1) = 1$. Обидва класи функцій, що зберігають нуль і одиницю є замкненими, що перевіряється підстановкою констант у суперпозиції. Функція f є самодвоїста, якщо $f(x_1,\ldots,x_n) = \overline{f}(\bar{x}_1,\ldots,\bar{x}_n)$. Клас самоподвійних функцій замкнений.

Теорема. (Друга основна теорема про функціональну повноту). Для того щоб система функцій Σ була функціонально повною у сильному змісті, необхідно і достатньо, щоб вона містила:

1) нелінійну функцію;
2) немонотонну функцію;
3) не самодвоїсту функцію;
4) функцію, що не зберігає 0;
5) функцію, що не зберігає 1.

Необхідність випливає із замкненості п'ятьох класів, що визначені в умові теореми.

Достатність. Під час доведення достатності слід відзначити, що оскільки леми використовують константи, то спочатку їх необхідно отримати (за умовами 3-5 даної теореми) і тільки потім застосувати попередню теорему.

Якщо $f(x_1,\ldots,x_n)$ не самодвоїста, то підстановкою в неї x і $\bar{x}$ можна одержати константу. Дійсно, з огляду на не самодвоїстість функції завжди знайдеться набір $(\sigma_1,\ldots,\sigma_n)$, такий, що $f(\overline{\sigma}_1,\ldots,\overline{\sigma}_n)=f(\sigma_1,\ldots,\sigma_n)$. Але тоді функція $f_c(x)=f(x^{\sigma_1},\ldots,x^{\sigma_n})$ є константою, тому що:

$$f_c(0)=f(0^{\sigma},\ldots,0^{\sigma})=f(\overline{\sigma}_1,\ldots,\overline{\sigma}_n)=f(\sigma_1,\ldots,\sigma_n)=f(1^{\sigma},\ldots,1^{\sigma})=f_c(1).$$

Нехай тепер f_0 не зберігає 0, f_1 - не зберігає 1, f_2 – не самодвоїста, f_0, f_1 і f_2 не обов'язково повинні різнитися.

Якщо $f_0(1,1,\ldots,1)=1$, то функція $\varphi(x)=f_0(x,\ldots,x)$ є константа 1 оскільки $\varphi(0)=1$ за визначенням f_0 і $\varphi(1)=f_0(1,\ldots,1)=1$, а функція $\psi(x)=f_1(\varphi(x),\ldots,\varphi(x))=f_1(1,\ldots,1)=0$, тобто $\psi(x)$ - константа 0.

Якщо ж $f_0(1,\ldots,1)=0$, то $\varphi(x)=\overline{x}$, тому що $\varphi(0)=1$ за визначенням f_0 і $\varphi(1)=f_0(1,\ldots,1)=0$. Але тоді із f_2 підстановкою x і $\overline{x}$ отримаємо функцію $f_c(x)$, що є константою, а використовуючи ще раз $\overline{x}$, одержимо другу константу. Таким чином, в будь-якому випадку виконання умов 3-5 даної теореми достатньо для одержання констант 0 і 1. Використовуючи цей факт і першу теорему про функціональну повноту, одержуємо другу основну теорему про функціональну повноту.

3 ТЕОРІЯ ГРАФІВ

Останнім часом теорія графів стала простим, доступним і надзвичайно ефективним інструментом для розв'язання задач, що відносяться до широкого кола проблем. Це проблеми проектування інтегральних схем і схем керування, дослідження автоматів, логічних кіл, блок-схем програм, економіки, статистики, теорії розкладання, а також дискретної оптимізації.

3.1 Теоретико-множинне введення

Це - не стільки введення, скільки нагадування деяких термінів. Усі вони розглядалися у попередніх частинах, однак нам вони будуть потрібні у дещо іншому розумінні. Отже, вихідні поняття: *множина і елемент.* Їх треба сприймати інтуїтивно, на рівні життєвого досвіду. Множина - це синонім слова сукупність. Множина складається з елементів. Якщо a - елемент множини A, то пишуть $a \in A$, а якщо a не є елементом множини A, то пишуть $a \notin A$. Символ $A = = \{a,b,c,\ldots\}$ означає, що множина A складається з елементів $a,b,c,\ldots$. Якщо потрібно символічно записати фразу «множина A складається з елементів a, що мають властивість f», то прийнято писати: $A=\{a|f\}$.

Символ $|A|$ позначає кількість елементів у множині A. Якщо спеціально не обговорено, то всі множини у наших розглядах будуть скінченними, тобто такими, що $|A|<\infty$.

Якщо кожен елемент множини A є елементом множини B, то говорять, що A - *підмножина* B і пишуть $A\subseteq B$. Прийнято спеціальним терміном і спеціальним символом позначати множини, що не містять жодного елемента. Їх називають *порожніми* множинами і позначають символом $\varnothing$. Якщо одночасно $A\subseteq B$ і $B\subseteq A$, то множини A і B називаються *рівними* і пишуть $A=B$. Над множинами можна здійснювати ряд традиційних дій:

1. **Об'єднання.** Це множина C, що включає в себе множини A і B. Позначення: $C=A\cup B$.

2. **Перетинання.** Це множина C, що будується за заданими множинами A і B таким чином, щоб вона містила всі елементи, що належать одночасно множині A і множині B. Позначення: $C=A\cap B$.

3. **Віднімання.** Це множина C, що будується за заданими множинами A і B таким чином, щоб вона містила всі елементи з A, які не належать множині B. Позначення: $C=A\setminus B$. Часто при включенні $A\subseteq B$ замість $A\setminus B$ пишуть $\overline{B}$ і говорять про *доповнення* підмножини B.

4. **Добуток.** Це множина C, що будується із заданих множин A і B, причому таким чином, щоб вона містила всі упорядковані пари (a,b), де $a\in A,\ b\in B$. Позначення: $C=A\times B$.

Якщо $A=B$, то $A\times A$ називається *декартовим квадратом* множини A. Підмножина елементів (a,a) у множині $A\times A$ називається *діагоналлю* множини A і позначається як $\Delta(A)$.

Нагадаємо ще декілька означень.

Відношення на множинах. Якщо в декартовому квадраті $A\times A$ деякої множини A виділена будь-яка підмножина X, то говорять що на A *задане відношення* (або *бінарне відношення*). Якщо для деяких елементів $u,v\in A$ має місце включення $(u,v)\in X$, то говорять, що u,v *знаходяться у відношенні* X.

Якщо $\Delta(A)\subseteq X\subseteq A\times A$, то відношення X називають *рефлексивним.*

Якщо відношення $U\subseteq A\times A$ таке, що з включення $(x,y)\in U$ обов'язково випливає $(y,x)\in U$, то відношення U називають *симетричним.*

Якщо відношення $U\subseteq A\times A$ таке, що з включень $(x,y)\in U$ і $(y,z)\in U$ випливає, що $(x,z)\in U$, то відношення U називають *транзитивним.*

Якщо відношення $U\subseteq A\times A$ таке, що з одночасних включень $(x,y)\in U$ і $(y,x)\in U$ випливає, що $x=y$, то відношення називається *антисиметричним.*

За визначенням розуміють, що порожня множина є відношенням одночасно рефлексивним, симетричним, антисиметричним і транзитивним.

Якщо відношення рефлексивне, транзитивне і симетричне, то воно називається *еквівалентністю*. Якщо ж відношення одночасно рефлексивне, транзитивне і антисиметричне, то воно називається *частковим порядком*.

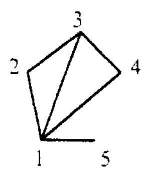
Якщо $U \subseteq A \times A$ - частковий порядок на множини A і $(x, y) \in A$, то говорять, що елементи x, y *порівнянні щодо* U або просто *порівнянні;* іноді при цьому пишуть $x \underset{U}{\leq} y$ або просто $x \leq y$.

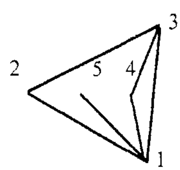
3.2 Означення графа. Суміжність й інцидентність

Означення графа. Нехай A - будь-яка множина. Позначимо через $V(A)$ множину усіх *неупорядкованих* пар його різних елементів. Наприклад, якщо A={1,2,3}, то $V(A)$={(1,2), (1,3), (2,3)}; якщо A={1,2}, то $V(A)$={(1,2)}. Якщо A={1}, то $V(A) = \varnothing$, оскільки різних елементів у A немає. Коли у записі $V(A)$={(1,2), (1,3), (2,3)} вказується пара (1,2), то розуміють, що вирази (1,2) і (2,1) означають одне й те саме: це означає, що пара *неупорядкована*, тобто не має значення, в якому порядку записані елементи пари. *Графом* називається пара множин $\Gamma = [A, B]$, де A - будь-яка не порожня множина, а $B \subseteq V(A)$. Елементи множини A називаються *вершинами* графа, а елементи множини B - його *ребрами*. Наприклад:

$$A = \{1,2,3,4,5\}, \quad B = \{(1,2),(2,3),(3,4),(1,5),(1,4),(3,1)\}.$$

Наведемо традиційну *геометричну інтерпретацію* графа. Нехай $\Gamma = [A,B]$ - деякий граф, при цьому, $A = \{a_1, a_2, ..., a_p\}$, $B = \{b_1, b_2, ..., b_q\}$. Зафіксуємо на площині довільним чином p точок і довільно дамо їм імена вершин. В результаті на площині виникнуть точки, позначені як $a_1, a_2, ..., a_p$. Потім для кожної пари точок a_i, a_j таких, що $(a_i, a_j) \in B$, проведемо відрізок прямої, що з'єднує точки a_i, a_j. В результаті таких дій виникне деякий рисунок, що і називається *геометричною інтерпретацією* графа. Слід відзначити, що одному і тому ж графові може відповідати багато рисунків, що можуть бути його геометричними інтерпретаціями. Нижче ілюструються приклади, що є геометричною інтерпретацією графа, наведеного вище як приклад:





Суміжність і інцидентність. Якщо у деякому графі $\Gamma = [A, B]$, де $A = \{a_1, a_2, ..., a_p\}$, $B = \{b_1, b_2 ... b_q\}$, пари вершин a_i, a_j такі, що $(a_i, a_j) \in B$, то вершини a_i, a_j називають *суміжними*. При цьому кожна з них називається інцидентною *ребру* (a_i, a_j), а ребро (a_i, a_j) називається інцидентним кожній з вершин a_i, a_j. Якщо вершина a_i і ребро b_j інцидентні, то пишуть $a_i \in b_j$.

Щоб описати граф $\Gamma = [A, B]$, де $A = \{a_1, a_2, ..., a_p\}$ - множина його вершини, побудуємо квадратну матрицю $M = (m_{ij})$, $i, j = 1, ..., p$, поклавши

$$m_{ij} = \begin{cases} 1, & (a_i, a_j) \in B, \\ 0, & (a_i, a_j) \notin B. \end{cases}$$

Очевидно, що ця матриця симетрична. Вона називається *матрицею суміжності* графа $\Gamma = [A, B]$. У наведеному вище прикладі графа матриця суміжності така:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Очевидно, що для неорієнтованого графа матриця m_{ij}=m_{ji} і всі його ребра визначаються верхньою трикутною матрицею. Кількість їх дорівнює $\sum_{i=1}^{p} \sum_{j=1}^{p} m_{ij}$ (тобто це кількість ребер у графі). Ребра орієнтованого графа визначаються всіма елементами m_{ij} матриці суміжності. За матрицею суміжності легко будується список ребер, що визначає граф.

Опишемо граф $\Gamma = [A, B]$ ще однією матрицею $N = (n_{ij})$, $i = 1, ..., p$, $j = 1, ..., q$, яку визначимо таким чином:

$$n_{ij} = \begin{cases} 1, & a_i \in b_j. \\ 0, & a_i \notin b_j. \end{cases}$$

Описана таким чином матриця N називається *матрицею інцидентності* даного графа. Очевидно, що вигляд матриці суміжностей і вигляд матриці інциденцій істотно залежать від того, як саме занумеровані вершини і ребра. Якщо в наведеному вище прикладі графа вважати, що

$$b_1 = (1,2),\ b_2 = (1,3),\ b_3 = (1,4),\ b_4 = (1,5),\ b_5 = (2,3),\ b_6 = (3,4),$$

то матриця інцидентності буде такою:

$$N = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Відзначимо, що в кожному стовпці матриці інцидентності графа тільки два елементи відмінні від 0 (або один, якщо ребро є петлею).

3.3 Степені вершин графа

Кількість ребер, інциденетних даній вершині a називається її степенем або *локальним степенем графа у вершині a*; степінь вершини a позначається як $d(a)$. У наведеному вище прикладі степінь вершини «1» дорівнює чотирьом, степінь вершини «2» дорівнює двом, степінь вершини «3» дорівнює трьом, степінь вершини «4» дорівнює двом, степінь вершини «5» дорівнює одиниці.

Наведемо приклад графа з локальним степенем нуль, що описується як: $A = \{1,2,3\}$, $B = \{(1,2)\}$. Тут вершина «3» має степінь 0. Вершини із степенем 0 називаються *ізольованими*. Можна перевірити, що *в будь-якому графі число вершин непарного степеня обов'язково парне*.

Локальні степені усіх вершин графа легко можна отримати з матриці інцидентності і матриці суміжності відповідно

$$\rho\ (v_j)=\sum_{i=1}^{q} n_{ij}\ ;\ \rho(v_j)=\sum_{j=1}^{p} m_{ij}\ .$$

Під час підрахунку степеня вершин за наведеними виразами кожна петля додасть одиницю у степінь відповідної вершини. Однак, оскільки під час зображення петлі до цієї вершини примикають два кінці петлі, степінь цієї вершини повинна збільшитись на два. З огляду на це, формулу розрахунку через коефіцієнти матриці інцидентності треба змінити таким чином:

$$\rho'(v_j)=\sum_{i=1}^{q} n_{ij}\,(\ 3\text{-}\sum_{k=1}^{q} n_{ik}).$$

Коли *i-те* ребро звичайне, то $\sum_{k=1}^{q} n_{ik} = 2$ і відповідний додаток зовнішньої суми n_{ij}, тобто одиниця для ребер, інцидентних вершині v_j і нуль для тих, що лишились. Якщо ж ребро є петлею, то $\sum_{k=1}^{q} n_{ik} = 1$, а додаток зовнішньої суми дорівнює $2n_{ij}$, тобто два для петель, інцидентних вершині v_j і нуль для тих, що лишились.

Ці ж значення степеня визначаються і через коефіцієнти матриці суміжності графа, причому відповідно до формули:

$$\rho'(v_j) = \sum_{i=1}^{p} m_{ij} + \delta_{jj} = \sum_{k=1}^{p} m_{jk} + m_{jj}.$$

Оскільки кожне ребро має два кінці, то в сумі $\sum_{v \in G} \rho'(v)$ ребра враховуються два рази. Таким чином ця сума дорівнює подвоєному числу ребер графа тобто вона парна. Отже, парна і кількість непарних доданків цієї суми тобто число вершин непарного степеня.

Граф називається *однорідним степеня k*, якщо степені всіх його вершин рівні *k*. Якщо однорідний граф степеня *k* має *n* вершин і *m* ребер, то

$$m = \frac{1}{2} \sum_{v \in G} \rho'(v) = kn/2,\ n = 2m/k.$$

Граф без кратних ребер називається повним, якщо кожна пара вершин з'єднана ребром.

3.4 Орієнтовані графи

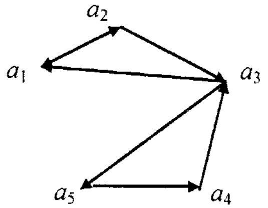
Орієнтований граф (скорочено *орграф*) - це пари множин $\overline{\Gamma} = [A, B]$, де A - будь-яка непорожня скінченна множина, а B - підмножина в $A \times A \setminus \Delta(A)$, де $A \times A$ - прямий добуток множини A на себе; $\Delta(A)$ - діагональ множини A. Елементи з A називаються *вершинами* орграфа, а елементи з B - *ребрами*.

Якщо $x, y \in A$ і $(x, y) \in B$, то x називається *початком*, а y називається *кінцем* ребра (x, y); як і в «неорієнтованому випадку», вершини x, y називаються *інцидентними* ребру (x, y), а ребро (x, y) називається *інцидентним* вершинам x, y.

Орграф має природну геометричну інтерпретацію: його вершини зображуються у вигляді точок на площині, а ребра – у вигляді стрілок з початку в кінець ребра. Наприклад, якщо орграф $\overline{\Gamma}=[A,B]$ має множину вершин $A=\{a_1,a_2,a_3,a_4,a_5\}$ і множину ребер

$$B=\{(a_1,a_2),(a_2,a_3),(a_2,a_1),(a_3,a_5),(a_4,a_3),(a_3,a_1),(a_5,a_4)\},$$

то геометрично він виглядає таким чином:



Якщо два орграфа $\overline{\Gamma}_1=[A_1,B_1]$, $\overline{\Gamma}_2=[A_2,B_2]$ такі, що одночасно виконуються дві умови - $A_2\subseteq A_1$ і $B_2\subseteq B_1$, - то говорять, що $\overline{\Gamma}_2$ - *підграф* орграфа $\overline{\Gamma}_1$. При цьому пишуть: $\overline{\Gamma}_2\subseteq\overline{\Gamma}_1$. Якщо одночасно виконуються умови $\overline{\Gamma}_2\subseteq\overline{\Gamma}_1$ і $\overline{\Gamma}_1\subseteq\overline{\Gamma}_2$, то орграфи $\overline{\Gamma}_1=[A_1,B_1]$, $\overline{\Gamma}_2=[A_2,B_2]$ називаються *рівними* і пишуть $\overline{\Gamma}_1=\overline{\Gamma}_2$.

Для вершин *орієнтованого* графа визначаються два локальних степеня: $\rho_1(v)$- число ребер з початком у вершині v; $\rho_2(v)$- число ребер з кінцем у вершині v; Петля дає внесок 1 в обидві сторони. Локальні степені вершин орієнтованого графа просто визначаються через коефіцієнти матриці суміжності:

$$\rho_1(v_i)=\sum_{j=1}^{q}n_{ij},\ \rho_2(v_i)=\sum_{k=1}^{q}n_{ki}.$$

при цьому визначення степеня вершин через коефіцієнти матриці інцидентності значно складніші.

Оскільки кожне ребро орієнтованого графа G має один початок і один кінець, то суми $\sum_{v\in G}\rho_1(v)$ і $\sum_{v\in G}\rho_2(v)$ рівні кількості ребер цього графа, а значить рівні між собою. Звідси випливає, що в однорідному орієнтованому графі степеня k з n вершинами і m ребрами:

$$m=\sum_{v\in G}\rho_1\ (v)=\sum_{v\in G}\rho_2\ (v)=k\,n,\ n=m/k.$$

Кожен орграф $\overline{\Gamma}=[A,B]$ описується *матрицею суміжностей*: якщо $A=\{a_1,...,a_p\}$, то побудуємо матрицю $M=(m_{ij})$, $i,j=1,...,p$, поклавши

$$m_{ij} = \begin{cases} 1, & якщо \quad (a_i, a_j) \in B, \\ 0, & якщо \quad (a_i, a_j) \notin B. \end{cases}$$

З кожним орграфом $\overline{\Gamma} = [A, B]$ однозначно зв'язується звичайний граф $\Gamma = [A, C]$, у якому множина вершин та ж сама (A), а множина ребер C виходить «стиранням стрілок» у ребер з множини B (тобто якщо $x, y \in A$, то $(x, y) \in C$ тоді і тільки тоді, коли або $(x, y) \in B$, або $(y, x) \in B$). Граф $\Gamma = [A, C]$ називається *асоційованим з орграфом* $\overline{\Gamma} = [A, B]$.

3.5 Частини, суграфи і підграфи

Граф H називається частиною графа G, $H \subset G$, якщо множина його вершин $V(H) \subset V(G)$ і ребер $E(H) \subset E(G)$.

Якщо $V(H) = V(G)$, то частина графа називається суграфом. Суграф H покриває вершини неорієнтованого графа G, якщо будь-яка вершина G інцидентна хоча б одному ребру з H. Таким чином якщо в графі G є ізольована вершина v, що не інцидентна жодному ребру, то суграфи цього графа, що покривають, не існують.

Підграфом $G(U)$ графа G з множиною вершин $U \subset V$ називається частина графа, якій належать усі ребра з обома кінцями з U.

Операції з частинами графа. Доповнення $\overline{H}$ частини H визначається множиною всіх ребер графа G, що не належать H.

Об'єднання $H1 \cup H2$ і перетинання $H1 \cap H2$ частин $H1$ і $H2$ графа G визначаються природним чином:

$$V(H1 \cup H2) = V(H1) \cup V(H2);$$
$$E(H1 \cup H2) = E(H1) \cup E(H2);$$
$$V(H1 \cap H2) = V(H1) \cap V(H2);$$
$$E(H1 \cap H2) = E(H1) \cap V(H2).$$

Дві частини графа $H1$ і $H2$ не перетинаються по вершинах, якщо вони не мають загальних вершин, а значить і загальних ребер. Об'єднання частин, що не перетинаються називаються прямою сумою. Дві частини графа не перетинаються по ребрах, якщо $E(H1 \cap H2) = \varnothing$. Наприклад, об'єднання частини графа і її доповнення є прямою сумою по ребрах.

3.6 Графи і бінарні відношення

Бінарне відношення може бути описане у вигляді графа з множиною вершин V: $(R) = G(V)$, так що ребро (для орієнтованого графа) $(a,b) \in G \Leftrightarrow aRb$. Нуль-граф відповідає нульовому відношенню $a\ 0\ b$, що не виконується ні для якої пари. Повний граф U відповідає універсальному відношенню

$a \cup b$, що виконується для будь-якої пари .

Оскільки кожне відношення R має додаткове відношення або заперечення $\rceil R$, то aRb істинне тільки тоді, коли aRb не виконується. Наприклад для відношення $a = b$ відношенням заперечення $\rceil R$ є $a \neq b$.

Граф $G(\rceil R)$ є *доповненням* графа $G(R)$ по відношенню до повного графа $U(V)$ з множиною вершин V, на якому задані бінарне відношення R і множина ребер $E(U(V)) = V \times V$.

Для будь-якого відношення R є *обернене* відношення R^{-1}, для якого на відміну від графа $G(R)$ напрямки всіх ребер змінені на протилежні. Граф $G(R^{1})$ є *оберненим* графу $G(R)$. Відношення $R = R^{-1}$, тобто $aRb \Leftrightarrow bRa$ називаються *симетричними* (їм відповідають неорієнтовані графи).

Говорять, що з відношення R випливає відношення R', або R' містить R. $R' \supset R$, якщо $aRb \rightarrow aR'b$. Відповідні графи $G(R)$ і $G(R')$мають одну і ту саму множину вершин, а множина ребер першого графа є підмножиною ребер другого. Таким чином $G(R)$ є *суграфом* $G(R')$, тобто $G(R') \supset G(R)$.

Для будь-яких двох відношень R_1 і R_2, які задані на одній множині вершин графа можна визначити об'єднання $R_1 \cup R_2$ і перетинання $R_1 \cap R_2$:

$$a(R_1 \cup R_2)b = aR_1b \cup aR_2b;$$
$$a(R_1 \cap R_2)b = aR_1b \cap aR_2b.$$

Відповідні графи також є об'єднанням і перетинанням:

$$G(R_1 \cap R_2) = G(R_1) \cap G(R_2);$$
$$G(R_1 \cup R_2) = G(R_1) \cup G(R_2).$$

Рефлективність - aRa $\forall a \in V$ (це означає, що в кожній вершині граф має петлю). Обернене відношення також буде рефлексивне. Відношення антирефлексивне, якщо aRa ніколи не виконується (граф не має петель), наприклад $a \neq b$ - антирефлексивне.

Транзитивність - aRb і $bRc \rightarrow aRc$. Для графа це означає, що якщо $G(R)$ містить ребра (a,b) і (b,c), то він також містить (a,c).

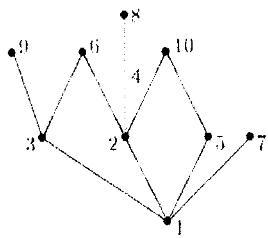
Відношення еквівалентності рефлексивне, симетричне і транзитивне. Всі елементи з V, еквівалентні даному елементу a, утворюють множини $R(a)$, які називаються класом еквівалентності елемента a. З рефлексивності випливає, що $a \in R(a)$. Якщо aRb і bRc, то із транзитивності випливає aRc, так що $R(a) \supset R(b)$. Тому із симетричності маємо, що $R(a) = R(b)$ при aRb.

Два різних класи еквівалентності $R(a)$ і $R(b)$ не можуть мати будь-який загальний елемент b, інакше $R(a) = R(b) = R(c)$, що не можливо. Таким чином, класи еквівалентності утворюють розбиття V, тобто розбиття на підмножини, що не перетинаються. Клас еквівалентності $R(a) = a$, називається особливим, інакше, не особливим.

Припустимо, що задано розбиття $V = \cap B_к$. Визначимо відношення еквівалентності R із цими класами $B_к$, вважаючи $aRb \Leftrightarrow a,b \in B_к$. У відповідному графі $G(R)$ будь-які дві вершини з однієї множини B_k будуть сполучені ребром, ніяке ребро не з'єднує вершини з різних множин. Отже $G(R) = \cup \cup_0 (B_k)$. Цей вираз є прямою сумою повних графів, визначених на різноманітних множинах B_k.

Часткове упорядкування. Відношення $a \geq b$ називається частково упорядкованим або відношенням вмикання, якщо: $a \geq b$ (рефлексивне); з $a \geq b$ і $b \geq a$ випливає $a = b$ (антисиметричність); транзитивність. Відповідний граф транзитивний, має петлі і будь-які дві вершини в ньому з'єднані одним ребром. Відношення вмикання називається відношенням упорядкування, а відповідна множина V – упорядкованою, якщо виконується ще така додаткова умова $\forall a,b \in V$ і виконується одне із співвідношень $a \geq b$, $a \leq b$.

До відношень часткового порядку відносяться також відношення ієрархії. Як приклад наведемо граф відношення ділимості. Побудуємо граф, що зображує відношення ділимості на множині {1,2,3,4,5,6,7,8,9,10}. Принцип такий: якщо від одного числа до іншого є ланцюг, що веде нагору, тоді друге число ділиться на перше.



Строге часткове впорядкування. Відношення $a > b$ називається строгим частковим упорядкуванням, або строгим вмиканням, якщо воно задовольняє умови: $a > b$ і $b > a$ не мають місця одночасно; транзитивність.

Очевидно, що строге часткове впорядкування це перетин часткового впорядкування і відношення $a \neq b$. Граф суворого часткового впорядкування утворюється з частково впорядкованого графа видаленням усіх петель.

3.7 Ізоморфізм графів

Ізоморфізм графів – це одне з найважливіших понять у теорії графів. Нехай $\Gamma_1 = [A_1, B_1]$, $\Gamma_2 = [A_2, B_2]$ - два графа. Припустимо, що існує таке відображення множини вершин $f: A_1 \to A_2$, що виконуються такі чотири умови:

- якщо $x, y \in A_1$, $x \neq y$, то $f(x) \neq f(y)$;

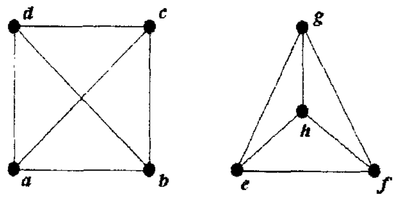
- для будь-якого $y \in A_2$ існує $x \in A_1$ таке, що $f(x) = y$;
- якщо $(x, y) \in B_1$, то $(f(x), f(y)) \in B_2$;
- $\forall\ (u, v) \in B_2$ існує таке $(x, y) \in B_1$, що $u = f(x)$ і $v = f(y)$.

Тоді відображення f називається *ізоморфізмом* графів Γ_1, Γ_2, а самі ці графи називаються *ізоморфними*. Неважко помітити, що для ізоморфізму кожна вершина переходить у вершину з тим же степенем. Тому напевно неізоморфними є графи у списку локальних степенів яких є різкі відмінності (наприклад, в одному графі є вершина зі степенем 3, а в іншому такого степеня взагалі немає). Однак, перевірка двох графів на ізоморфізм є набагато складнішою задачею, ніж просте порівняння степенів.

Для орієнтованих графів два орграфа - $\overline{\Gamma}_1 = [A_1, B_1]$, $\overline{\Gamma}_2 = [A_2, B_2]$ - називаються *ізоморфними*, якщо існує відображення $f : A_1 \to A_2$ таке, що виконуються такі чотири умови:

- якщо $x, y \in A_1$ і $x \neq y$, то $f(x) \neq f(y)$;
- для будь-якого $v \in A_2$ існує таке $u \in A_1$, що $f(u) = v$;
- якщо $(x, y) \in B_1$, то $(f(x), f(y)) \in B_2$;
- якщо $(u, v) \in B_2$ і $u = f(x)$, $v = f(y)$, то $(x, y) \in B_1$.

Поняття ізоморфізму для графів має наочне тлумачення. Уявимо ребра графів еластичними нитками, що зв'язують вузли – вершини. Тоді, ізоморфізм можна уявити як переміщення вузлів і розтягання ниток. Для прикладу покажемо, що наведені нижче два графи є ізоморфними.



Дійсно, відображення $a \to e$, $b \to f$, $c \to g$, $d \to h$, що є ізоморфізмом, легко уявити як модифікацію наведеного на рисунку зліва графа, що пересуває вершину d у центр рисунку.

3. 8 Маршрути, ланцюги, цикли. Стандартні операції над графами

Нехай G - неорієнтований граф. *Маршрутом* у графі G називається така скінченна або нескінченна послідовність ребер $S = (\ldots\ e_0,\ e_1, \ldots\ , e_n, \ldots)$, що кожні два сусідніх ребра e_{i-1} і e_i мають загальну інцидентну їм вершину. Тобто можна написати $\ldots e_0 = (a_0, a_1)$, $e_1 = (a_1, a_2)$, ... , $e_n = (a_n, a_{n+1})$, ...

Відзначимо, що одне і те саме ребро може повторюватися. Вершина a_0, інцидентна e_1 і не інцидентна e_2 називається початковою. Аналогічно визначається кінець маршруту.

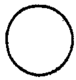
Вершини, які інцидентні ребрам маршруту, окрім початкової і кінцевої, називаються *внутрішніми* або *проміжними*.

Якщо маршрут s скінченний, то його можна записати як $S = (a_0, a_n)$ де a_0, a_n- кінцеві вершини маршруту. Кількість ребер маршруту називається його *довжиною*.

Якщо $a_0 = a_n$, то маршрут називається *циклічним*.

Маршрут називається *ланцюгом*, а циклічний маршрут - *циклом*, якщо кожне його ребро зустрічається в ньому не більше одного разу (вершини і ланцюги можуть повторюватися декілька разів). Цикл, у якому немає повторюваних вершин, крім кінцевих, називається *простим*.

Нижче наведено схематичне зображення простого циклу (ліворуч) і схематичне зображення не простого циклу (праворуч):



Наведені вище поняття відносяться також і до орієнтованих графів. У кожному орграфі $\overline{\Gamma} = [A, B]$ виділяється основний об'єкт - *орієнтований шлях* (коротко - *оршлях*). *Оршлях* - це символ

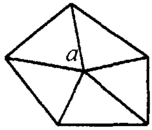
$$x_1(x_1, x_2)x_2(x_2, x_3)x_3 \ldots x_k(x_k, x_{k+1})x_{k+1} \ldots x_{n-1}(x_{n-1}, x_n)x_n \ ,$$

у якому $x_i \in A$, $i = 1, \ldots, n$ (причому, у списку $x_1, \ldots x_n$ можуть бути повторювані вершини) і в який $(x_i, x_{i+1}) \in B$, $i = 1, \ldots, n-1$. Ясно, що під час переходу від орієнтованого графа до асоційованого з ним графа, оршлях відтворює конкретний звичайний шлях. У наведених вище позначеннях вершина x_1 називається *початком* оршляху, а вершина x_n називається *кінцем* оршляху. При цьому про сам оршлях говорять, що *він є оршляхом* з x_1 у x_n.

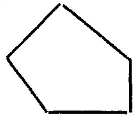
Як і раніше, для орієнтованих графів уводяться поняття ланцюга, простого ланцюга, циклу і простого циклу. Ці об'єкти можна досліджувати таким самим чином, що і для неорієнтованих графів.

Стандартні операції над графами. До таких операцій у загальному випадку відносять операції видалення вершин графа, видалення ребер графа і розбиття ребер.

Видалення вершини. Нехай $\Gamma = [A, B]$ - граф і $a \in A$. Видалити вершину a із графа Γ - це значить побудувати новий граф $\Gamma' = [A', B']$, у якому $A' = A \setminus \{a\}$ і B' виходить з B видаленням усіх ребер, що інцидентні вершині. Нижче ілюструється видалення вершини a з графа:

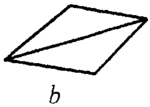


до видалення вершини a

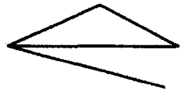


після видалення вершини a

Видалення ребра. Нехай $\Gamma = [A,B]$ - граф і $b \in B$. Видалити ребро b графа - це значить побудувати новий граф $\Gamma' = [A',B']$, у якому $A' = A$ і $B' = B \setminus \{b\}$. Нижче ілюструється видалення ребра графа:

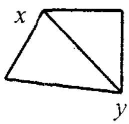


до видалення ребра b

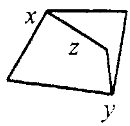


після видалення ребра b

Розбиття ребер. Нехай $\Gamma = [A,B]$ - граф і $b = (x,y) \in B$. Розбити ребро b - це значить побудувати новий граф $\Gamma' = [A',B']$, у якому $A' = A \cup \{z\}$ (тобто z - деяка нова вершина) і $B' = (B \setminus \{b\}) \cup \{(x,z),(z,y)\}$. З графічної точки зору ця операція означає «внесення в ребро нової вершини». Нижче ілюструється операція розбиття ребра (x,y) графа:



до розбиття ребра



після розбиття ребра

3.9 Найкоротші маршрути

Граф $\Gamma = [A,B]$ називається *зваженим*, якщо визначена будь-яка функція $f : B \to \Re$ (функція на множині ребер із значеннями на множині дійсних чисел). Сама функція f називається *ваговою*, а її значення на тому або іншому ребрі називається *вагою* цього ребра. Будь-який підграф даного графа і будь-який маршрут у даному графі *мають свою вагу*: це - сума ваг ребер, що входять у цей підграф або в цей маршрут. Якщо граф не зважений, то можна вважати ваги дуг рівними одиниці.

На зваженому графі $\Gamma = [A,B]$ з ваговою функцією f можна ввести поняття *відстані* між вершинами. Робиться це так. Будемо вважати далі, якщо не буде спеціальних застережень, що всі розглянуті графи зв'язні. Вагу будь-якого маршруту будемо називати *довжиною* цього маршруту. Ясно, що між двома вершинами існує такий маршрут, що має мінімальну можливу довжину. Ця довжина і називається *відстанню* між двома вершинами. Маршрут, який цю довжину реалізує, називається *найкоротшим*. Наприклад, для графа на рис. 3.1 нижче можливі маршрути між вершинами $x_н=x_1$ і $x_к=x_6$ можуть бути такими:

I. $(x_1x_2)\,(x_2x_4)\,(x_4x_6)$; $\ell=1+5+4=10$.
II. $(x_1x_3)\,(x_3x_5)\,(x_5x_6)$ $\ell=4+4+2=10$
III. $(x_1x_2)\,(x_2x_3)\,(x_3x_5)\,(x_5x_6)$ $\ell=1+2+4+2=9$ і.т.д.

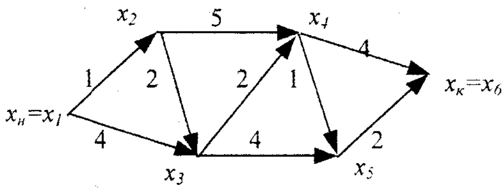


Рисунок 3.1

Очевидно, що з переглянутих маршрутів найменшу довжину $\ell=9$ має третій. Однак чи є він найкоротшим? Звичайно, для такого невеликого графа можна ще перебирати маршрути і вибрати найкоротший, але для великих і складних графів потрібен метод, який скорочує перебір.

Задача про найкоротший маршрут в графі має багато інтерпретацій: відстань між геометричними точками; час реалізації програми між виділеними мітками; вартість передачі інформації каналами зв'язку між заданими пунктами в мережі зв'язку, і т.п.

Хвильовий алгоритм побудови найкоротшого маршруту у незваженому графі. Для незваженого графа довжина маршруту – це число дуг, які входять в нього. Алгоритм побудови найкоротшого маршруту оснований на покроковому розповсюдженні хвилі від вершини початку маршруту по графу до тих пір, поки хвиля не торкнеться вершини кінця маршруту, і потім виділяються дуги, по яких хвиля дійшла до кінцевої вершини.

0. Вершина $x_н$ помічається як 0, а інші вважаються не поміченими; $i=0$.

1. $i=i+1$. Помічаються індексом i всі непомічені раніше вершини, в яких є дуги від помічених вершин. Якщо помічена вершина $x_к$, то виконується п.2, інакше, якщо поточним значенням індексу i виявились поміченими будь-які вершини, то виконується п.1, інакше робиться висновок, що маршрутів з вершини $x_н$ в вершину $x_к$ немає.

2. Зворотним проходом по дугах, починаючи від вершини $x_к$, виділяються ті дуги і вершини, які інцидентні виділеним вершинам і різниця між

вагами яких рівна 1. Під час руху від вершини $x_н$ по відзначених дугах виявляються побудовані всі найкоротші маршрути до вершини $x_к$ (рис. 3.2).

Приклад. Вершина x_1 помічена i=0, вершини x_2 і x_3 помічаються i=1, потім вершини x_4 і x_5 – помічаються i=2 і, нарешті, вершина x_6=$x_к$ помічається i=3. Зворотній прохід: вершина x_6 виділена, різниці ваг пар вершин x_4,x_6 і x_5,x_6=1, виділяються дуги (x_4,x_6) і (x_5,x_6) і вершини x_4 і x_5. Від цих виділених вершин відповідно їх входам обчислюються різниці ваг пар вершин x_4,x_2, x_4,x_3 і x_5,x_3, які дорівнюють одиниці, і виділяються дуги (x_2,x_4), (x_3,x_4) і (x_3,x_5) і вершини x_2 і x_3. Кінцева різниця ваг пар вершин x_2,x_1 і x_3,x_4 дорівнює 1, виділяються дуги (x_1,x_2) і (x_1,x_3). Рухаючись від вершини x_1 по виділених дугах, отримуємо три найкоротші маршрути:

I. (x_1,x_2), (x_2,x_4), (x_4,x_6);
II. (x_1,x_3), (x_3,x_4), (x_4,x_6);
III. (x_1,x_3), (x_3,x_5), (x_5,x_6).

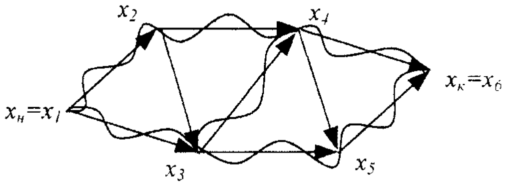


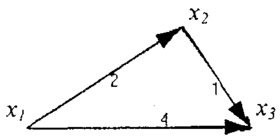
Рисунок 3.2

Довжини ℓ всіх маршрутів однакові і дорівнюють числу ***i*** кроків розповсюдження хвилі (ℓ=i=3).

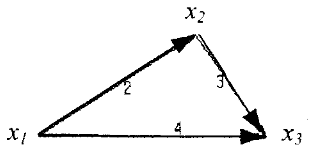
Хвильовий алгоритм для зваженого графа. Відзначимо особливості побудови найкоротшого маршруту для зваженого графа за хвильовим алгоритмом.

I. Замість одиничної зміни індекса i (номера кроків розповсюдження хвилі) під час обчислення пройденого маршруту для кожної вершини повинна використовуватися сума ваг ℓ_{ij} дуг (x_i, x_j), які ведуть до цієї вершини від вершини $x_н$. До кожної вершини можуть йти від вершини $x_н$ декілька маршрутів, суми довжин дуг по різних маршрутах відмінні. Під час пошуку найкоротшого маршруту необхідно вибирати найменшу суму. Хвилі розповсюдження ваги по різних маршрутах доходять до кожної вершини послідовно, при черговій хвилі необхідно або залишити стару вагу вершини, або замінити її на нову (меншу). Тому під час розрахунку ваги вершини x_i за рахунок хвилі, що дійшла до неї по дузі (x_j, x_i), обчислення ваги V_i визначається за формулою $V_i = min(V_i, V_j+\ell_{ij})$.

Приклади.



Крок 0: $V_1 = 0$.
Крок 1: $V_2 = 0 + 2 = 2$,
$V_3 = 0 + 4 = 4$.
Крок 2: $V_3 = \min(4, 2 + 1) = 3$.



Крок 0: V_1=0.
Крок 1: $V_2 = 0 + 2 = 2$,
$V_3 = 0 + 4 = 4$.
Крок 2: $V_3 = \min(4, 2 + 3) = 4$.

2. Ваги вершин в процесі розповсюдження хвилі можуть змінюватися неодноразово. При кожній зміні ваги V_i вершини це зменшення необхідно передати вершинам виходу x_i, тобто необхідні спеціальні засоби, які відображують факти отримання вершиною нової ваги і передачу її іншим вершинам. В якості такого засобу використовуються масиви номерів вершин, які отримали нову вагу (під час кожної зміни ваг номери вершини включаються у цей масив, якщо їх там не було, а під час передачі ваг – виключаються з нього).

Формулювання алгоритму

1. Вершина $x_н$ отримує вагу $V_н = 0$, її номер вводиться в масив M номерів вершин, які змінили вагу. Всі інші вершини x_i, що лишилися, отримують вагу $V_i = \infty$ і їх номери не попадають в масив M.

2. Якщо масив M порожній, то виконується п.3, інакше вибирається з виключенням з нього чергова вершина x_i і перераховуються ваги вершин, що належать виходу $G(x_i)$ вершини x_i: $\forall x_i \in G(x_i)$ $(V_j = \min(V_j, V_i + \ell_{ij}))$.

Якщо вага V_j зменшується, то номер j включається зі зведенням подібних в M. Далі знову виконується п.2.

3. Якщо вага $V_к = \infty$, то робиться висновок, що маршрутів з вершини $x_н$ до вершини $x_к$ немає, інакше виконується процедура виділення дуг, така ж, як і в хвильовому алгоритмі для незваженого графа, за винятком того, що різниця ваг вершин x_i і x_j повинна бути рівною ℓ_{ij}. Після виділення дуг будуються найкоротші маршрути, довжини яких дорівнюють $V_к$. Розглянемо рис. 3.1.

1: $V_н = V_1 = 0$, $V_2 = V_3 = V_4 = V_5 = V_6 = \infty$, $M = \{1\}$.
2: $M \neq \varnothing$, $i = 1$, $M = \varnothing$, $G(x_1) = \{x_2 x_3\}$;
$V_2 = \min(\infty, 0+1) = 1$, $M = \{2\}$;
$V_3 = \min(\infty, 0+4) = 4$, $M = \{2,3\}$.

2: $M \neq \varnothing$, $i = 2$, $M = \{3\}$, $G(x_2) = \{x_3 x_4\}$;
$V_3 = \min(4, 1+2) = 3$, $M = \{3\}$;
$V_4 = \min(\infty, 1+5) = 6$, $M = \{3,4\}$.

2: $M = \varnothing$, $i = 3$, $M = \{4\}$, $G(x_3) = \{x_4 x_5\}$;
$V_4 = \min(6, 3+2) = 5$, $M = \{4\}$;
$V_5 = \min(\infty, 3+4) = 7$, $M = \{4,5\}$.
2: $M = \varnothing$, $i = 4$, M=$\{5\}$, $G(x_4) = \{x_5 x_6\}$;
$V_5 = \min(7, 5+1) = 6$, $M = \{5\}$;

$V_6 = \min(\infty, 5+4) = 9$, $M=\{5,6\}$.

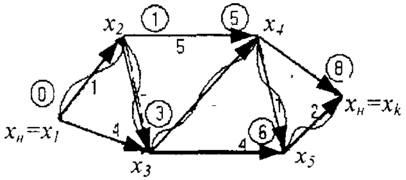
2: $M \neq \varnothing$, $i = 5$, $M = 6$, $G(x_5) = \{x_6\}$;

$V_6 = \min(9, 6+2) = 8$, $M = \{6\}$;

2: $M \neq \varnothing$, $i = 6$, $M = \varnothing$, $G(x_6) = \varnothing$;

2: $M = \varnothing$ і виконується п.3.

Для ілюстрації виконання п.3. наведемо граф (див. рис. 3.1) з вагами вершин, отриманими в результаті розв'язання прикладу на рис. 3.2.



Починаємо з вершини x_6.

$G^{-1}(x_6) = \{x_4, x_5\}$; $(\ell_{4,6} = 4) \neq (V_6 - V_4 = 3)$, тому дуга (x_4, x_6) не виділяється;

$(\ell_{5,6} = 2) = (V_6 - V_5 = 2)$ виділяється дуга (x_5, x_6) і вершина x_5.

Для виділеної вершини x_5: $G^{-1}(x_5) = \{x_3, x_4\}$;

$(\ell_{3,5} = 4) \neq (V_5 - V_3 = 3)$, дуга (x_3, x_5) не виділяється;

$(\ell_{4,5} = 1) = (V_5 - V_4 = 1)$, виділені дуга (x_4, x_5) і вершина x_4.

Для вершини x_4: $G^{-1}(x_4) = \{x_2, x_3\}$;

$(\ell_{2,4} = 5) \neq (V_4 - V_2 = 4)$, дуга (x_2, x_4) не виділяється;

$(\ell_{3,4} = 2) = (V_4 - V_3 = 2)$, виділяється дуга (x_3, x_4) і вершина x_3.

Для вершини x_3: $G^{-1}(x_3) = \{x_1, x_2\}$;

$(\ell_{1,3} = 4) \neq (V_3 - V_1 = 3)$, тому дуга (x_1, x_3) не виділяється;

$(\ell_{2,3} = 2) = (V_3 - V_2 = 2)$ виділяється дуга (x_2, x_3) і вершина x_2.

Для вершини x_2: $G^{-1}(x_2) = \{x_1\}$;

$(\ell_{1,2} = 1) = (V_2 - V_1 = 1)$, виділяється дуга (x_1, x_2).

У результаті побудовано найкоротший маршрут $(x_н, x_к)$ довжиною 8: $(x_н=(x_1, x_2)(x_2, x_3)(x_3, x_4)(x_4, x_5)(x_5, x_6=x_к)$. Процес розв'язання можна описати у вигляді такої таблиці:

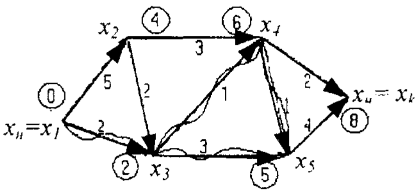
x_1	x_2	x_3	x_4	x_5	x_6	M
0	∞	∞	∞	∞	∞	1
0	1	4	∞	∞	∞	2,3
0	1	3	6	∞	∞	3,4
0	1	3	5	7	∞	4,5
0	1	3	5	6	9	5,6
0	1	3	5	6	8	6
0	1	2	5	6	8	∅

Далі процес розв'язання виконується за рис. 3.2.

Наведемо ще один приклад (див на рисунку нижче) з коротким записом процесу розв'язання. На наведеному графі відмітимо вершини отриманими числовими значеннями ваг і виділимо найкоротші маршрути. Отримано два найкоротших маршрути довжиною 8:

1. $(x_1, x_3)(x_3, x_2)(x_2, x_5)(x_5, x_4)(x_4, x_6)$;
2. $(x_1, x_3)(x_3, x_5)(x_5, x_4)(x_4, x_6)$.

x_1	x_2	x_3	x_4	x_5	x_6	M
0	∞	∞	∞	∞	∞	1
0	5	2	∞	∞	∞	2,3
0	5	2	8	6	∞	3,4,5
0	4	2	8	5	∞	4,5,2
0	4	2	8	5	10	5,2,5
0	4	2	6	5	9	2,6,4
0	4	2	6	5	9	6,4
0	4	2	6	5	9	4
0	4	2	6	5	8	4
0	4	2	6	5	8	∅



3.10 Алгоритм Дейкстри

Як видно з останнього прикладу, ваги вершин V_i за попереднім алгоритмом неодноразово перераховувались і з'являлись декілька раз в масиві M, будучи джерелом нових хвиль розповсюдження ваг. Алгоритм Дейкстри дозволяє усунути цей недолік. Його відмінність полягає в тому, що на кожному кроці обирається для розповсюдження ваги вершина, що має найменшу вагу, яка не розповсюдила вагу на інші вершини. Тобто, в алгоритмі Дейкстри відмічаються вершини, що вже розповсюдили свою вагу, а з інших вибираються чергові верщини з найменшою вагою.

Формулювання алгоритму.

1. $V_н = 0$, $\forall x_i (i \neq H)(V_i = \infty)$. Всі вершини не відмічені.

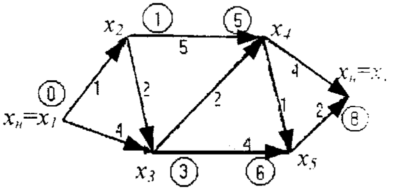
2. Серед невідмічених вершин вибираємо x_i з найменшою вагою V_i і відмічаємо її. Якщо $V_i = \infty$, то робимо висновок, що маршрутів з вершини $x_н$ до вершини $x_к$ немає, інакше, якщо вибрана вершина $x_i = x_к$, то виконуємо п.3, інакше перераховуємо ваги вершин, що належать $G(x_i)$: $V_j = \min(V_j, V_i+\ell_{ij})$. Далі знову виконуємо п.2.

3. Аналогічний п.3 хвильового алгоритму.

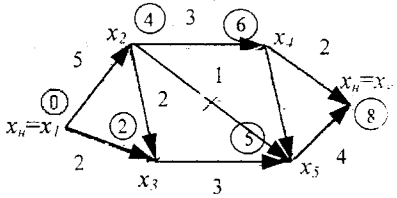
Наведемо скорочений запис розв'язання за алгоритмом Дейкати для прикладів, що наведені на рис. 3.1 і 3.2, розглянуті вершини позначаються у кружечках. Найкоротші маршрути не виділяємо, оскільки цей процес виконується так, як і у хвильовому алгоритмі.

Як бачимо, процес розв'язання для обох прикладів трохи скоротився, хоча на кожному кроці необхідно вибирати вершину меншої ваги.

x_1	x_2	x_3	x_4	x_5	x_6
0	∞	∞	∞	∞	∞
	1	4	∞	∞	∞
		3	6	∞	∞
			5	7	∞
				6	9
					8



x_1	x_2	x_3	x_4	x_5	x_6
0	∞	∞	∞	∞	∞
	5	2	∞	∞	∞
	4		∞	5	∞
			7	5	∞
			6		9
					8



Примітка про найдовші маршрути. В багатьох додатках поряд з найкоротшими маршрутами постає задача побудови найдовших маршрутів. Наприклад, в програмі необхідно знати не тільки найменший, але й найбільший час її реалізації. Для розв'язання таких задач достатньо в алгоритмах пошуку найкоротшого маршруту розрахунок ваг вершин здійснювати відповідно до правила $V_j = max(V_j, V_i+\ell_{ij})$, а у першому пункті хвильового алгоритму присвоїти всім вершинам вагу 0.

Під час побудови найдовших маршрутів розглядаються *елементарні* або прості найдовші маршрути, найдовші маршрути *із заданим числом виконання циклу.*

Приклад. Розв'язання задачі для графу на рис. 3.1 буде мати вигляд:

x_1	x_2	x_3	x_4	x_5	x_6	M
0	0	0	0	0	0	1
0	1	4	0	0	0	<u>2</u>,3
0	1	4	6	0	0	<u>3</u>,4
0	1	4	6	8	0	<u>4</u>,5
0	1	4	6	8	10	<u>5</u>,6
0	1	4	6	8	10	<u>6</u>
0	1	4	6	8	10	∅

У результаті побудовано три найдовші маршрути довжиною 10:

1. $(x_1, x_2)(x_2, x_4)(x_4, x_6)$;
4. $(x_1, x_3)(x_3, x_4)(x_4, x_6)$;
3. $(x_1, x_3)(x_3, x_5)(x_5, x_6)$.

3.11 Алгоритм Форда для пошуку найкоротших маршрутів

Класичним алгоритмом пошуку найкоротших маршрутів у графі є *алгоритм Форда*. Оснований він на такому простому факті: *якщо є найкоротший маршрут між будь-якими двома вершинами, то його частина між будь-якими двома вершинами на ньому ж самому також є найкоротшим маршрутом*. Алгоритм Форда дозволяє знайти найкоротші маршрути з будь-якої однієї вершини графа в усі інші. Цю вихідну вершину можна вибрати довільно, однак, зробивши вибір, далі уже треба виходити з того, що будуть описані найкоротші маршрути саме з цієї вершини в усі інші.

Опишемо даний алгоритм по кроках.

Крок 0. Пронумеруємо усі вершини з $\Gamma = [A, B]$, так що $A = \{1, 2, ..., p\}$ і при цьому номер «1» має саме та вершина, з якої будуть знайдені найкоротші маршрути в усі інші вершини.

Побудуємо далі матрицю $M = (m_{ij}),\quad i, j = 1, 2, ..., p$, поклавши

$$m_{ij} = \begin{cases} f((i,j)), \quad \textit{якщо} \quad (i,j) \in B, \\ \textit{клітина лишається незаповненою,} \quad \textit{якщо} \quad (i,j) \notin B. \end{cases}$$

Крок 1. Біля першого рядка матриці M, ліворуч від матриці, поставимо числову позначку «0» і таку ж позначку проставимо над першим стовпцем матриці. Потім переглянемо позначений рядок зліва направо і кожен раз, зустрічаючи клітинку з числом, додамо це число до позначки рядка, а суму проставимо над стовпцем, у якому ця клітинка знаходиться.

Далі відобразимо позначки стовпців щодо головної діагоналі. Виникнуть позначені рядки. Для кожного з позначених рядків проробимо те же саме: переглянемо позначений рядок зліва праворуч і кожен раз, зустрічаючи клітинку з числом, додамо це число до позначки рядка, а суму поставимо як позначку над стовпчиком, у якому ця клітинка міститься. При цьому будемо дотримуватися принципу $\wp$: «наявну позначку не змінювати». Потім позначки стовпців відобразимо щодо головної діагоналі і з позначеними рядками знову проробимо те ж саме. І так далі, поки не виявляться позначеними всі рядки і всі стовпці.

Крок 2. Переглянемо рядки таблиці в порядку зростання їх номерів. У кожному рядку проглядаються клітинки зліва направо і кожен раз, коли зустрічається число, воно додається до позначки рядка, а сума порівнюється з позначкою стовпця, у якому знайдене число розташоване. Якщо сума виявилася менше, ніж позначка стовпця, то ця позначка стовпця заміняється на згадану суму. Якщо ж сума виявилася більша або рівна позначці, то нічого не змінюється. Після такого перегляду всіх рядків нові позначки стовпців відображаються щодо головної діагоналі і вся процедура повторюється. І так доти, поки не припиняться зміни в позначках.

Крок 3. Тепер за позначками можна побудувати найкоротші маршрути з першої вершини в усі інші. Фіксуємо довільну вершину k $(k = 2,3,\ldots,p)$ й опишемо найкоротший маршрут з першої вершини у вершину k. По-перше, довжина цього найкоротшого маршруту дорівнює позначці λ_k, що міститься над стовпцем з номером k. По-друге, *передостання* вершина в найкоротшому маршруту з першої вершини у вершину k знаходиться таким чином: у стовпці з номером k відшукуємо число, сума якого з позначкою рядка, у якому воно розташовано, дорівнює λ_k. Нехай номер рядка, у якому знайдене число виявилося, дорівнює l. Тоді передостанньою вершиною в найкоротшому маршруті з 1 у k буде вершина l. Вершину, що передує вершині l, треба відшукувати як передостанню в найкоротшому маршруті з 1 в l і так далі.

Наведемо приклад. Нехай вихідний зважений граф описується матрицею M, що наведена нижче:

	10	13					
10						11	
13			17	10			
		17			16	12	
					11		15
			16	11		15	10
	11		12		15		
				15	10		

Розставимо позначки:

	0	10	13	30	23	46	21	38
0		10	13					
10	10						11	
13	13			17	10			
30			17			16	12	
23						11		15
46				16	11		15	10
21		11		12		15		
38					15	10		

Проведемо зменшення позначок:

		0	10	13	30	23	34	21	38
		0	10	13	30	23	46	21	38
0	0		10	13					
10	10	10						11	
13	13	13			17	10			
30	30			17			16	12	
23	23						11		15
34	46				16	11		15	10
21	21		11		12		15		
38	38					15	10		

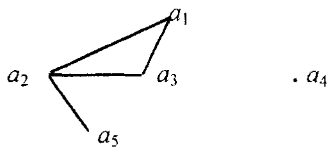
Вкажемо найкоротші маршрути:

1⇒2: довжина 10; маршрут (від кінця до початку) 2⇐1;

1⇒3: довжина 13; маршрут (від кінця до початку) 3⇐1;
1⇒4: довжина 30; маршрут (від кінця до початку) 4⇐ 3⇐1;
1⇒5: довжина 23; маршрут (від кінця до початку) 5⇐3⇐1;
1⇒6: довжина 34; маршрут (від кінця до початку) 6⇐7⇐2⇐1;
1⇒7: довжина 21; маршрут (від кінця до початку) 7⇐2⇐1;
1⇒8: довжина 38; маршрут (від кінця до початку) 8⇐5⇐3⇐1.

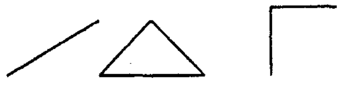
3.12 Зв'язні графи. Компоненти зв'язності

Нехай G - неорієнтований граф. Дві вершини a і b називаються *зв'язними*, якщо існує маршрут S з початком a і кінцем b. Наприклад, у графі



вершини a_3 і a_5 є зв'язними за маршрутом $a_3(a_3,a_2)a_2(a_2,a_5)a_5$, а вершини a_4 і a_1 не зв'язні ні за яким маршрутом. Граф називається *зв'язним*, якщо будь-яка пара вершин зв'язна. Таким чином, наведений приклад графа є незв'язним.

У загальному випадку відношення зв'язності для вершин графа є відношенням еквівалентності. Звідси випливає те, що існує розкладання загальної множини вершин $V=\bigcup_i V_i$, причому таким чином, що $V_i \cap V_j = \varnothing$, $i \neq j$, тобто вершини однієї підмножини V_i є зв'язними, а вершини з різних підмножин V_i не зв'язні. Звідси випливає розкладання $G=\bigcup_i G(V_i)$ графа G на зв'язні підграфи, які не перетинаються, $G(V_i)$. Підграфи $G(V_i)$ називаються *компонентами зв'язності* графа G. Можна запропонувати ще й таке визначення. *Зв'язним компонентом* графа називається такий його підграф, що є сам по собі зв'язним графом, який при цьому збігається з будь-яким іншим, що містить його зв'язним підграфом. Таким чином, зв'язний граф має єдиний зв'язний компонент - це він сам. На рисунку нижче наведено приклад графа з трьома зв'язними компонентами:



В орієнтованих графах вершини x_i і x_j називаються *зв'язними слабо* якщо в графі (G,X) існує тільки маршрут (x_i, x_j). Якщо ж в графі (G,X) існують маршрути (x_i, x_j) і (x_j, x_i), то вершини x_i і x_j будуть *зв'язні сильно*. Якщо в графі немає маршрутів з вершини x_i в x_j і нема зворотного маршруту з x_j в x_i, то вершини x_i і x_j не зв'язні.

Так само, як і для неорієнтованих графів, поняття зв'язності приводить до поняття зв'язного компонента: підграф $\overline{\Gamma}_1$ орграфа $\overline{\Gamma}$ називається *зв'язним компонентом* орграфа $\overline{\Gamma}$, якщо: 1) $\overline{\Gamma}_1$ є зв'язним орграфом; 2) не існує зв'язного орграфа $\overline{T}$ такого, що $\overline{\Gamma}_1 \subseteq \overline{T} \subseteq \overline{\Gamma}$ і $\overline{\Gamma}_1 \neq \overline{T}$.

Очевидно, що для неорієнтованого графа має сенс лише поняття сильної зв'язності. Відношення сильної зв'язності рефлексивне, симетричне і транзитивне, тобто воно є відношенням еквівалентності і однозначно розбиває множину вершин графа на компоненти зв'язності: максимальні підмножини сильно зв'язних між собою вершини. На рис. 3.3 наведено приклад графа, що має три компоненти зв'язності:

1 - $\{x_1, x_2, x_3, x_4\}$;
2 - $\{x_5, x_6, x_7\}$;
3 - $\{x_8\}$.

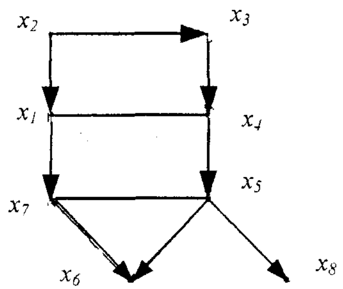


Рисунок 3.3

При цьому легко перевірити, що всередині кожної компоненти між всіма парами вершин є маршрути в обох напрямках. Між компонентами – тільки слабка зв'язність: є маршрути з вершин компоненти 1 в вершини компонент 2 і 3, і з вершини компоненти 2 в вершину x_8.

Побудова компонент зв'язності у неорієнтованому графі. Як випливає з однозначності розбиття неорієнтованого графа на компоненти зв'язності їх визначення не потребує перебору варіантів. Тому алгоритм побудови компонент зв'язності досить простий і складається всього з двох пунктів.

1: $i = 0$. Всі вершини графа не відмічені.

2: $i = i+1$. Вибираємо наступну вершину, відмічаємо її і всі зв'язані з нею вершини значенням індексу i за допомогою розповсюдження хвилі по ребрам, що виходять від уже відмічених індексами вершин. Таким чином, виділяється i-та компонента зв'язності. Якщо є ще не відмічені вершини, то виконується п.2, інакше визначення компонент зв'язності завершено.

Розглянемо описаний вище алгоритм для графа, що наведений на рисунку нижче.

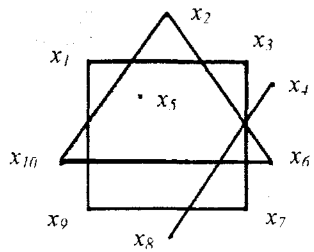
1: $i = 0$.

2: $i = 1$. Відмічаємо індексом $i = 1$ вершину x_1 і зв'язані з нею вершини x_3, x_7, x_9. Отримана перша компонента зв'язності: $\{x_1, x_3, x_7, x_9\}$.

2: $i = 2$. Відмічаємо індексом $i = 2$ вершину x_2 і вершини x_6 і x_{10}. Побудована друга компонента зв'язності.

2: $i = 3$. Відмічаються індексом $i = 3$ вершини x_4 і x_8. Третя компонента зв'язності - $\{x_4, x_8\}$.

2: $i = 4$. Відмічається індексом $i = 4$ вершина x_5, яка формує четверту компоненту зв'язності – $\{x_5\}$



Побудова компонент зв'язності в орієнтованому графі. Алгоритм полягає в перетворенні вихідного орієнтованого графа до неорієнтованого маршрутом перетворення, не порушуючи сильної зв'язності вершин графа. Тому будемо мати на увазі, що для отриманого неорієнтованого графа алгоритм визначення компонент зв'язності вже сформульований в попередньому підрозділі.

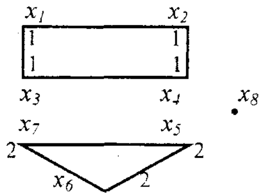
Алгоритм перетворення орієнтованого графа до неорієнтованого.

1. Всі зустрічні дуги в орієнтованому графі замінюємо ребрами.

2. Якщо в перетворюваному графі уже нема дуг, то процедура закінчена, інакше вибирається будь-яка дуга (x_i, x_j) і, якщо в графі є маршрут (x_j, x_i), то така дуга (x_i, x_j) і всі дуги маршруту (x_j, x_i) замінюються ребрами. Якщо ж маршрутів (x_j, x_i) в графі немає, то дуга (x_i, x_j) видаляється з графа, тобто повинна зберігатися тільки сильна зв'язність. Виконується п. 2.

Покажемо процедуру визначення за даним алгоритмом компонент зв'язності для графа на рис. 3.3. Для дуги (x_2, x_3) є маршрут (x_3, x_4) (x_4, x_1) (x_1, x_2), рівний маршруту (x_3, x_2), тому дуги (x_2, x_3) і (x_3, x_4) заміняємо ребрами. Для дуги (x_4, x_5) маршрутів (x_5, x_4) в графі немає, тому дуга (x_4, x_5) видаляється. Аналогічно видаляються дуги (x_1, x_7) і (x_5, x_8). Для дуги (x_5, x_6) є маршрут (x_6, x_5), рівний маршруту (x_6, x_7) (x_7, x_5), тому дуги (x_5, x_6) і (x_6, x_7) заміняються ребрами.

Таким чином, отримано неорієнтований граф (див. на рисунку нижче), визначення компонент зв'язності для якого не викликає труднощів за попереднім алгоритмом.



Наведемо кілька теорем щодо зв'язності графів.

Теорема. Кожний неорієнтований граф розпадається єдиним чином у пряму суму своїх зв'язних компонент.

Доповнення $\bar{G}$ графа G у повному графі з тією ж множиною вершин також мають єдине розбиття на свої зв'язні компоненти.

Теорема. Якщо в скінченному графі G рівно дві вершини a_0 і b_0 мають непарний локальний степінь, то вони зв'язні.

Доведення. Оскільки в скінченному графі число вершин непарного степеня парне, то це твердження виконується і для тієї компоненти графа G, до якої належить вершина a_0. Тоді і вершина b_0 повинна належати тій самій компоненті. Якщо H - частина графа G, в якій усі локальні степені непарні, то в графі $\neg H$ (який утворюється з G видаленням H; $\neg H = G \setminus H$) a_0 і b_0 повинні залишитися зв'язними.

Теорема. Нехай G є зв'язним графом і H є його частиною. Тоді число зв'язних компонент доповнення $\neg H$ графа H у G не перевершує числа вершин у H.

Теорема. Якщо граф G з однократними ребрами і без петель має n вершин і k зв'язних компонент, то максимальне число ребер у G дорівнює

$$N(n,k)=\frac{1}{2}(n-k)(n-k+1)$$

Теорема. Граф з n вершинами і з числом ребер, більшим, ніж $N(n,2)=(n-1)(n-2)/2$ є зв'язним.

3.13 Взаємно-однозначні відображення графів. Відстань, діаметр, радіус і центр графів

Взаємно-однозначні відображення графів (ВОВ). Розглянемо граф, обумовлений ВОВ, або підстановкою $a \to a' = \tau(a)$, $a \in V$ елементів множини V. Таке відображення можна інтерпретувати як бінарне відношення $a \tau a'$, що виконується тільки тоді, коли a' є поданням a при відображенні τ.

У графі G відображення τ в кожній вершині a буде єдине вихідне ребро $(a, \tau(a))$ і єдине вхідне ребро $(a, \tau^{-1}(a))$. Тоді для локальних степенів ми маємо $\rho(a)=\rho^{-1}(a)=1$, тобто G є однорідний граф степеня 1.

Обернення будь-якого однорідного орієнтованого графа степеня 1 визначає ВОВ τ множини V на себе, що утворюється за формулою $a \leftrightharpoons \tau(a)$ для кожного орієнтованого ребра (a, a'). Для ВОВ τ кожна вершина a визначає єдиний орієнтований маршрут $a, \tau(a), \tau^2(a) \ldots$. Тут можливі два випадки.

1.В маршруті існують повторювані вершини, тобто таке ціле число m і n, для яких можна записати, що $\tau^m(a)=\tau^{m+n}(a)$. Оскільки кожна вершина є поданням єдиної вершини, отримаємо $\tau^n(a)=a$. Найменше $n>0$ із цією властивістю називається порядком a при відображенні τ. Звідси можна зробити висновок, що маршрут складається з ребер деякого простого циклу і подає орієнтований простий цикл довжиною n, який проходить через вершини $G_n(a)=(a, \tau(a), \tau^2(a), \ldots, \tau^{n-1}(a))$. Цей цикл є одним із зв'язних компонент.

2.Всі вершини в маршруті різні. Оскільки a - єдине подання для $\tau^1(a)$, $\tau^{-1}(a)$, $\tau^{-2}(a)$ і т.д., то маршрут можна розглядати як двосторонню нескінченну послідовність вершин:

$$G_\infty(a)=(\ldots \tau^{-2}(a), \tau^{-1}(a), a, \tau(a), \tau^2(a), \ldots).$$

Говорять, що a має нескінеченний порядок щодо τ і належить нескінченному простому орієнтованому циклу.

Теорема. ВОВ множин V на себе визначається однорідними орієнтованими графами степеня 1 на V. Пов'язані компоненти такого графа є або орієнтованими простими циклами, або двостороннє нескінченними орієнтованими простими ланцюгами.

Відстань. Нехай G - зв'язний неорієнтований граф $\forall a,b \in G$. Тоді існує простий ланцюг $S(a,b)$. Довжиною цих ланцюгів є цілі позитивні числа. Якщо кількість ребер цього ланцюга не мінімальна з можливих, то існує ланцюг, що має меншу кількість ребер. Отже, між a і b повинні існувати ланцюги найменшої довжини. Ця найменша довжина називається відстанню $d(a,b)$ між a і b. Вважаючи, кожну вершину неорієнтованого графа зв'язаною саму із собою, ми по суті вводимо нульові маршрути, що не мають ребер. Очевидно, що при цьому $d(a,a)=0$. Відстань $d(a,b)$ задовольняє аксіомам такої метрики:

1. $d(a,b) \geq 0$;
2. $d(a,b)=0 \Leftrightarrow a=b$;
3. $d(a,b)=d(b,a)$
4. $d(a,b)+d(b,c) \geq d(a,c)$ (нерівність трикутника).

Діаметр, радіус і центр графа. Нехай G - зв'язаний неорієнтований граф. Діаметр скінченного графа - це максимальна відстань між двома його вершинами $d(G) = \max\limits_{a,b\in V} d(a,b)$. Найкоротші прості ланцюги, що зв'язують дві вершини з максимальною відстанню, називаються *діаметральними простими ланцюгами*.

Нехай c - деяка довільна вершина. Максимальною відстанню графа від вершини c називається величина $r(G)=\max\limits_{x\in V} d(c,x)$. Вершина c_0 називається центром графа G, якщо величина $r(G) = \min\limits_{c_0\in V} r(c_0)$.

Максимальна відстань $r(G)$ від центра називається радіусом графа G, а будь-який найкоротший простий ланцюг від центра c_0 до будь-якої вершини, яка має максимальну відстань від c_0 називається радіальним ланцюгом. Центр не обов'язково повинен бути єдиним.

Протяжність. Нехай G скінченний зв'язаний неорієнтований граф, m – кількість його ребер. Кількість послідовностей цього графа без повторень дорівнює $m!$, тобто скінченна. Значить скінченне і число простих ланцюгів, де ребра не повторюються. *Протяжністю* між вершинами a і b $l(a,b)$ називається максимальна довжина найдовшого між ними простого ланцюга. Протяжність $l(a,b)$ також задовольняє аксіоми метрики.

У графі існують діаметральні за протяжністю або найдовші прості ланцюги. Їх довжина l_0 називається діаметром протяжності.

Для кожної вершини графа існують найдовші прості ланцюги з кінцем у цій вершині. Їх довжина $l(v)=\max\limits_{v\in V} l(v,x)$ називається *числом протяжності* для вершини v. *Центрами протяжності* називається вершина s_0 з мінімальним числом протяжності $l_0 = l(s_0) = \min\limits_{v\in V} l(v)$. Найдовші прості ланцюги від цих центрів називають *радіальними по протяжності* простими ланцюгами, а їх довжину l_0 - *радіусом протяжності*.

Теорема. Будь-які два найдовші прості ланцюги мають загальні вершини.

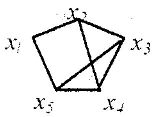
Теорема. Число протяжності задовольняє нерівності $l(v) \geq l_0/2$ або $l(v) \geq (l_0+1)/2$ відповідно для парного або непарного ланцюга. Рівність може бути досягнута коли v розташовано на кожному найдовшому простому ланцюзі.

Теорема. Якщо $\rho(a,b)$ - простий ланцюг із вершини a, що не можна продовжити за b, то b лежить на простому циклі, довжина якого не менша ніж $\rho(b)+1$.

Теорема. У скінченому графі G без петель і кратних ребер нехай l_0 довжина найдовшого простого ланцюга, і i_0 - максимальний індекс компонент по всіх простих ланцюгах. Тоді для локальних степенів у G існує верхня границя $\rho(v) \leq l(l_0, i_0), v \in V$, що залежить від G.

3.14 Система незалежних циклів

Нагадаємо, що замкнений маршрут в неорієнтованому графі називається циклом. В загальному випадку для графа можна побудувати декілька циклів. Наприклад, для наведеного нижче графа можна побудувати такі цикли:



I. $(x_1, x_2)\,(x_2, x_3)\,(x_3, x_4)\,(x_4, x_5)\,(x_5, x_1)$;
II. $(x_1, x_2)\,(x_2, x_4)\,(x_4, x_5)\,(x_5, x_1)$;
III. $(x_1, x_2)\,(x_2, x_3)\,(x_3, x_5)\,(x_5, x_1)$;
IV. $(x_1, x_2)\,(x_2, x_4)\,(x_4, x_3)\,(x_3, x_5)\,(x_5, x_1)$
і т.д.

Означення системи незалежних циклів. Цикл називається лінійно-залежним від деякої сукупності інших циклів, якщо його можна побудувати за допомогою лінійної комбінації циклів цієї сукупності. Щоб пояснити поняття лінійної комбінації, розглянемо сукупність двох циклів наведеного вище графа:

$$(x_1, x_2)\,(x_2, x_3)\,(x_3, x_5)\,(x_5, x_1);$$
$$(x_3, x_5)\,(x_5, x_4)\,(x_4, x_3)$$

і організуємо відповідно цим маршрутам їх обхід.

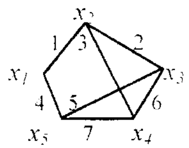
Виключаючи з обходу ребро (x_3, x_5), яке пройшло за цими маршрутами туди і назад, отримуємо новий цикл $(x_1,x_2)\,(x_2, x_3)\,(x_3, x_4)\,(x_4, x_5)\,(x_5, x_1)$. Цей цикл побудований як результат лінійної комбінації вихідних циклів і є лінійно-залежним від них. Таким чином, безпосередньо на графі неформально пояснено поняття лінійної залежності.

Дамо формальне означення. Кожному циклу поставимо у відповідність двійковий m-розрядний вектор, де m – число ребер графа. Пронумеруємо ребра. Для i-го циклу компонент C_{ij} вектора C_i визначається таким чином:

$$C_{ij} = \begin{cases} 0, & \text{якщо ребро } j \text{ не входить в цикл;} \\ 1, & \text{якщо ребро } j \text{ входить в цикл.} \end{cases}$$

Тоді лінійною комбінацією векторів C_i називається результат векторної операції додавання за модулем два. Для наведеного вище прикладу нумерація ребер графа показана нижче на рисунку.

1	2	3	4	5	6	7	
1	1	0	1	1	0	0	C_I
0	0	0	0	1	1	1	C_{II}
1	1	0	1	0	1	1	C_I+C_{II}



Таким чином *системою незалежних циклів графа* називається максимальна лінійно-незалежна сукупність циклів графа.

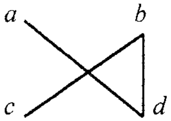
Цикломатичне число ν. Граф в загальному випадку може мати не одну систему незалежних циклів, проте число ν циклів в системі для кожного графа визначається його структурою однозначно. Число ν називається *цикломатичним числом графа*.

Теорема. Потужність ν системи незалежних циклів графа визначається формулою $\nu = m-n+p$, де m – число ребер; n – число вершин; p – число компонент зв'язності.

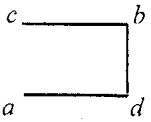
У наведеному вище прикладі ν =7-5+1=3, тобто в систему включаються три взаємно-незалежних цикли. Кожний наступний цикл вже можна побудувати лінійною комбінацією інших циклів.

3.15 Планарні і плоскі графи

Граф $\Gamma = [A, B]$ називається *планарним*, якщо хоча б одна з його геометричних інтерпретацій така, що ребра графа перетинаються тільки в його вершинах. Наприклад, граф



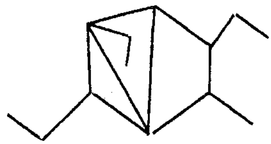
є планарним, оскільки його можна зобразити таким чином:



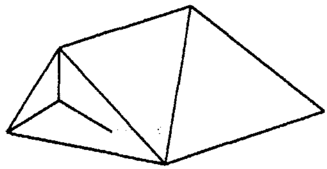
Планарний граф, що зображений на площині у такому вигляді, що його ребра перетинаються тільки в його вершинах, називається *плоским*. Таким чином, останній із зображених вище графів є плоским.

У математиці доводиться, що *не є планарними два таких графи:* K_5 (повний граф з п'яти вершин) і $K_{3,3}$ (повний дводольний граф, що має по три вершини). Існує класична *теорема Понтрягіна-Куратовського*, що стверджує таке: *граф планарний тоді і тільки тоді, коли в ньому немає підграфів, що отримані розбиттям ребер з* K_5 *і* $K_{3,3}$.

Опишемо класичну конструкцію Ейлера, що пов'язана з плоскими графами. Нехай $\Gamma = [A, B]$ - плоский граф. Це означає, що є певне зображення графа на площині, у якому ребра перетинаються тільки у вершинах. Кожен простий цикл такого графа обмежує дві частини площини: одна - внутрішня (обмежена), інша - зовнішня необмежена). Якщо по частині площини, обмеженій простим циклом плоского графа, не проходить жоден ланцюг цього графа початок і кінець якого належать обговорюваному циклу, то ця частина площини називається *гранню* плоского графа. Якщо грань необмежена, то вона називається *зовнішньою*. Обмежена грань плоского графа називається *внутрішньою*. Можна довести, що *в кожному плоскому графі обов'язково є зовнішня грань, причому тільки одна.* Підрахуємо для прикладу кількість граней плоского графа. Нехай маємо такий плоский граф:



Тут - чотири грані, причому одна з них - зовнішня. Відзначимо, до речі, що тут вершин - 13, а ребер - 15. Підрахуємо ті ж параметри (кількість граней, вершин і ребер) ще в одному прикладі:



Тут граней - 5, вершин - 7, ребер - 10.

Класична *теорема Ейлера* стверджує: *якщо в плоскому графі є «м» граней, «у» вершин і «р» ребер, то обов'язково виконується рівність: «м»* + *«у»* - *«р»*=2. Звідси виникає множина співвідношень для плоских графів, з яких зупинимося тільки на одному. Фіксуємо будь-який плоский граф. Нехай у ньому p вершин і q ребер. До цього графа можна, при необхідності, додати ребра і отримати знову ж таки плоский граф, з тією ж кількістю вершин, але із збільшеною кількістю ребер, причому такий, що всі грані в ньому будуть обмежені трикутниками (включаючи зовнішню). Якщо Q - нова кількість ребер і F - кількість граней у цьому новому графі, то $3F = 2Q$. Таким чином теорема Ейлера здобуває вигляд: $F + p - Q = 2$, але $3F = 2Q$, так що $2Q/3 + p - Q = 2$, звідки $Q = 3p - 6$, а *для довільного плоского графа*, отже, $q \leq 3p - 6$.

3.16 Дерево графа і його особливості

Зв'язний граф без циклів називається *деревом*. З властивостей зв'язності і відсутності циклів випливає, що число ребер в дереві на одиницю менше числа вершин. Є ще одна особливість дерев: будь-які дві вершини в дереві зв'язні, причому єдиним простим ланцюгом. Обидві ці обставини мають нескладні доведення, які пропонуємо довести самостійно.

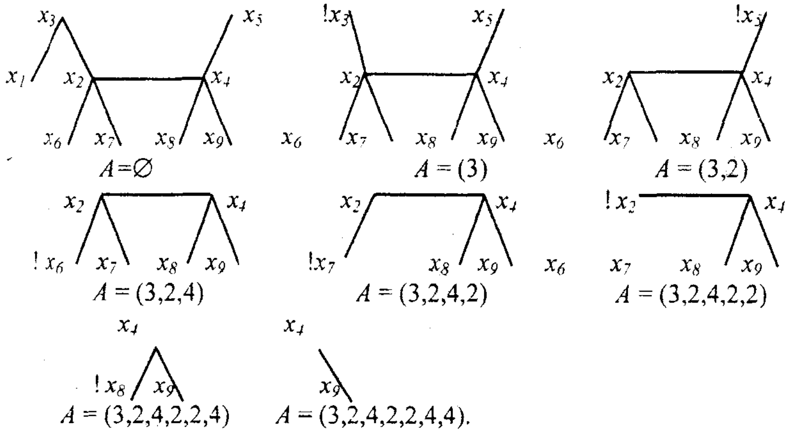
Вершини дерев, що інцидентні лише одному ребру, називаються *листками*, тобто вершинами, що висять. При цьому одна з вершин може бути оголошена коренем дерева.

Теорема. Існує $D_n=n^{n-2}$ різних дерев на n вершинах.

Доведення зводиться до двох конструктивних побудов: перетворення дерева до набору номерів вершин і, навпаки, до відновлення дерева по цьому набору номерів вершин. Однозначність процесу дозволяє припустити, що число дерев дорівнює числу наборів і таким чином розрахувати число дерев, уточнюючи властивості наборів. Процес побудови набору по дереву.

1. Розглядаємо дерево з $n \geq 2$ вершинами. Вихідний набір пустий.

2. Якщо перетворюване дерево зберегло лише дві вершини, то процес закінчений, інакше знаходимо серед листків вершину з найменшим номером. Цю вершину виключаємо з дерева разом з інцидентним їй ребром, в набір включаємо номер вершини з якою з'єднана вершина, що видаляється. Знову виконуємо п.2. Для прикладу дивись рисунок, що наведений нижче.

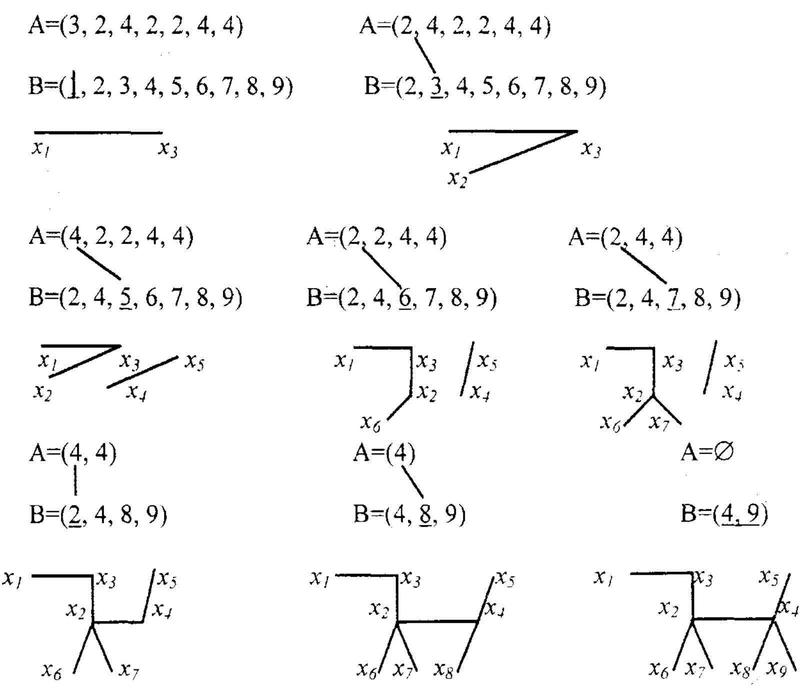


Сформулюємо процес відновлення дерева по набору A:

1. Будуємо набір B, рівний послідовності номерів вершин в порядку збільшення: $B=(1,2,...n)$.

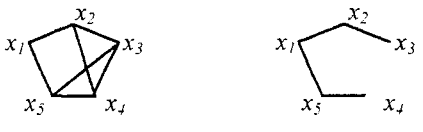
2. Якщо в наборі B залишилось всього два номери вершин, то з'єднаємо їх, тоді шукане дерево побудоване, інакше знаходимо в B мінімальний номер, якого немає в наборі A, і з'єднуємо цю вершину з вершиною, номер якої вказаний першим в наборі A. Викреслюємо ці номера з A і B. Знову виконуємо п.2.

Для даного прикладу процес відновлення показаний на рисунку нижче.



Виходячи з вище описаного $D_n=n^{n-2}$. Тобто формула доведена.

Визначення стовбура графа. Якщо граф не є зв'язним, то для визначення стовбурів слід розглядати його окремі компоненти зв'язності. При цьому стовбур графа у цілому буде визначатися як сукупність стовбурів окремих його компонент зв'язності. Для кожної ж компоненти зв'язності частковий підграф, що може бути побудований з неї видаленням деяких ребер, і який є деревом, називається *стовбуром*. У загальному випадку для графа можна побудувати декілька стовбурів. На рисунку нижче наведено приклад вихідного графа і один із можливих стовбурів цього графа.



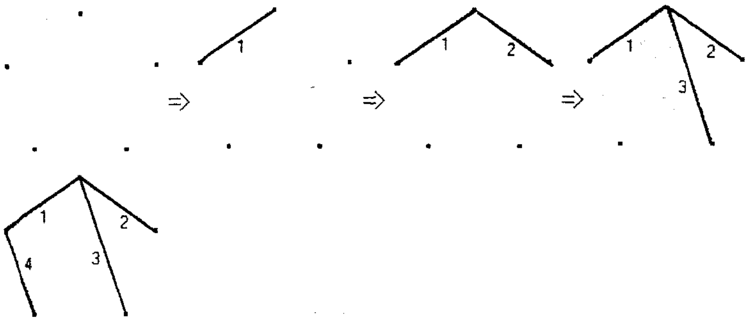
Алгоритм побудови довільного стовбура.

1. Для кожної компоненти i графа виконуємо п.2 і 3 алгоритму.
2. Будуємо частинний граф, що має всі n_i вершин компоненти і не

має ребер, тобто граф буде пустим (∅).

3. Якщо в поточний частинний граф включені уже n_i-1 ребер, то стовбур для компоненти ***і*** побудований, інакше вибираємо наступне нерозглянуте ребро компоненти і намагаємось включити його в поточний граф. Якщо у поточному графі це не приводить до створення циклу, то включаємо ребра, інакше – не включаємо. Ребро вважається розглянутим. Виконуємо п.3.

Наведемо приклад для розглянутого вище, в даному підрозділі, першого графа. В ньому є тільки одна компонента. На рисунку нижче зображена послідовність включення ребер, в розглянутому вище їх порядку.



Оскільки цикл не створився, то всі ребра з номерами 1,2,3,4 включені в стовбур і оскільки $m=n$-1, то стовбур вже побудований.

Якщо вибрати іншу послідовність перегляду ребер для включення в стовбур (наприклад, 5,6,7,2,3,1,4), то не всі розглянуті ребра будуть включені в стовбур, тому будується інший стовбур.

Побудова мінімального стовбура для зважених графів. Для зваженого графа стовбур з найменшою сумою ваг ребер, які в нього ввійшли, називається *мінімальним* (*найкоротший зв'язний стовбур*). Вище було показано, що під час розглядання ребер в сформульованому раніше алгоритмі побудови довільного стовбура в порядку зростання їх ваг буде побудований мінімальний стовбур.

Існує, також, *алгоритм Краскала*, що дозволяє знайти стовбур мінімальної ваги в будь-якому зваженому графі. Дамо його описання по кроках.

Крок 1. Знайдемо в даному графі ребро мінімальної ваги (якщо таких ребер декілька, фіксуємо кожне). Позначимо його через B_1. Крім того, фіксуємо підграф графа Γ, що складається з кінців ребра B_1 і самого цього ребра. Позначимо цей підграф через Γ_1.

Крок 2. Фіксуємо в вихідному графі Γ друге ребро - позначимо його через B_2, - вага якого мінімальна щодо ваг усіх ребер, що не належать Γ_1. Підграф, що складається з ребер B_1, B_2 і їхніх кінців позначимо через Γ_2.

Крок 3. Фіксуємо в графі Γ ребро - позначимо його через B_3, що має мінімальну вагу серед усіх ребер графа Γ, що не належать Γ_2. Підграф, що складається з ребер B_1, B_2, B_3 і їхніх кінців, позначимо через Γ_3.

Крок 4. Фіксуємо в графі Γ ребро - позначимо його через B_4, що має мінімальну вагу серед тих ребер графа Γ, що не належать Γ_3 і не утворюють циклу з ребрами з Γ_3. Підграф, що складається з ребер B_1, B_2, B_3, B_4 і їхніх кінців позначимо як Γ_4.

Загальний крок - крок № k. Фіксуємо в графі Γ ребро – позначимо його через B_k, що чає мінімальну вагу серед ребер, які не входять в Γ_{k-1} і не складають циклу з ребрами з Γ_{k-1}. Підграф, що складається з ребер B_1, B_2, B_3,..., B_k, позначимо через Γ_k.

Можна показати, що *якщо у вихідному графі Γ кількість вершин дорівнює p, то підграф Γ_{p-1} буде шуканим стовбуром.*

Задача про мінімальний стовбур має очевидні інтерпретації: ℓ_{ij} – довжина, тоді маємо найкоротшу зв'язність всіх вершин; ℓ_{ij} – вартість, найдешевша зв'язність всіх вершин, тощо.

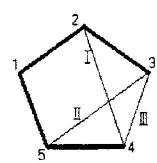
3.17 Алгоритм побудови системи незалежних циклів графа

1. Будується довільний стовбур графа. У вихідному графі відмічаються ребра, включені в стовбур.

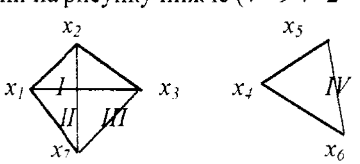
2. Вибирається наступне невідмічене ребро і будується цикл, що містить це ребро і ребра стовбура. Такий цикл обов'язково існує, оскільки вершини стовбура зв'язні. Розглянуте ребро відмічається і якщо є ще деяке ребро, то виконується п.2. Інакше виконується п.3.

3. За формулою Ейлера $\nu=m-n+p$ відбувається перевірка числа побудованих циклів (для контролю). Нижче наводиться приклад для наведеного праворуч графа.

Система циклів.
I. (x_2, x_4) (x_4, x_5) (x_5, x_1) (x_1, x_2);
II. (x_3, x_5) (x_5, x_1) (x_1, x_2) (x_2, x_3);
III. $(x_3, x_4)(x_4, x_5)(x_5, x_1)(x_1, x_2)(x_2, x_3)$;
ν=9-7+1=3.



Розглянемо більш складний приклад незв'язного графа, що наведений на рисунку нижче (ν=9-7+2=4).



Система циклів:
I. (x_1, x_3) (x_3, x_2) (x_2, x_1);
II. (x_2, x_7) (x_7, x_4) (x_1, x_2);
III. (x_3, x_7) (x_7, x_1) (x_1, x_2) (x_2, x_3);
IV. (x_5, x_6) (x_6, x_4) (x_4, x_5).

3.18 Розфарбування графа

Нехай $\Gamma = [A,B]$ і $f : A \to \mathrm{N}$ - будь-яка функція на множині вершин з натуральними значеннями. Така функція називається *розфарбуванням*, якщо її значення на будь-яких двох суміжних вершинах різні. Значення розфарбування, як функції, називаються *кольорами*, так що щодо будь-якого розфарбування суміжні вершини є різнокольоровими.

Існує давня задача про пошук мінімального числа кольорів, якими можна розфарбувати граф. У принципі, здійснити таке розфарбування графа можна прямим перебором можливих ситуацій за такою схемою.

Відзначимо, що якщо $A = \{a_1, a_2, ..., a_p\}$, то будь-яка функція на A з натуральними значеннями - це таблиця вигляду:

A	a_1	a_2	...	a_p
$f(A)$	n_1	n_2	...	n_p

Тут у першому рядку перераховані вершини графа, а в другому - ті натуральні числа, які дана функція відносить вершинам. Очевидно, що достатньо розглядати ті ситуації, у яких усі $n_1, n_2, ..., n_p \in \{1,2,...,p\}$. Отже, загальна кількість функцій вигляду $f : A \to \{1,2,...,p\}$ цілком обчислювальна - це p^p. Частина цих функцій є розфарбуванням, а частина - ні. Перевірити, чи є дана функція розфарбуванням, можна перебираючи всі ребра з B: з'ясовується чи однакові значення функції на кінцях ребер. Маючи повний список розфарбувань, можна знайти серед них такі, для яких кількість значень функції мінімальна.

Існує *теорема Ейлера про розфарбування плоских графів*: будь-який *плоский граф можна розфарбувати п'ятьма кольорами*. Не можна не відзначити, що реально поки що не знайдено жодного плоского графа, який не можна було б розфарбувати *чотирма* кольорами. Мінімальне число кольорів, якими можна розфарбувати граф $\Gamma = [A,B]$, називається *хроматичним числом* графа. Позначається ця характеристика символом $\chi(\Gamma)$. Знайти хроматичне число можна, зокрема, перебором усіх розфарбувань, про що говорилося вище.

Можна довести, що якщо $\Delta(\Gamma)$ позначає максимальний локальний степінь графа Г, то *справедлива нерівність*: $\chi(\Gamma) \le \Delta(\Gamma)+1$. При цьому важливо уявляти, що як рівність $\chi(\Gamma) = \Delta(\Gamma)+1$ реально можлива, наприклад, для графа, яким є дві з'єднані вершини, так і реально можлива нерівність $\chi(\Gamma) < \Delta(\Gamma)+1$, наприклад, для графів типу «зірки». При цьому для наведеного прикладу чим більше ребер у такій конструкції, тим більше $\Delta(\Gamma)$, а хроматичне число залишається рівним 2.

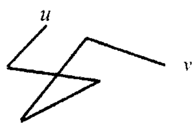
Будемо позначати символом $\pi(\Gamma,t)$ кількість розфарбувань графа Г за допомогою t кольорів, причому слова «за допомогою t кольорів» означають, що кольори «вибираються» з множини $\{1,2,...,t\}$. Ясно, що $\pi(\Gamma,t)$ - функція на множині натуральних чисел із значеннями на тій же множині. Ця функція називається *хроматичною функцією* даного графа.

Неважко помітити, що якщо $\Gamma = K_p$ - повний граф на p вершинах, то $\pi(\Gamma,t) = t(t-1)(t-2)...(t-p+1)$. Таким чином, для повних графів хроматична функція є багаточленом.

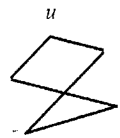
Існує теорема Зикова про хроматичні функції, завдяки якій виникає практична можливість побудувати хроматичну функцію у явному вигляді для будь-якого графа. Для формулювання цієї теореми потрібно попереднє роз'яснення одного терміна.

Фіксуємо в графі $\Gamma = [A,B]$ будь-які дві несуміжні вершини $u, v \in A$; слова «*стягнемо (чи склеїмо) ці вершини в одну вершину*» означають побудову нового графа $\Gamma' = [A',B']$, в якому $A' = A \setminus \{v\}$ й у якому B' складається з двох підмножин ребер: перша - це всі ребра з B, що неінцидентні ні u, ні v; друга - складається з усіх ребер, що інцидентні u, і всіх пар (u,x) таких, що $(v,x) \in B$. Геометрично сказане означає, що вершину v з усіма ребрами, що в неї входять, ніби приклеюють до вершини u, розтягуючи при цьому деякі ребра. Нижче наведено два приклади стягування вершин:

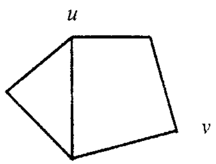
до стягування



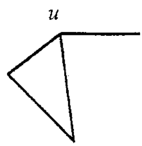
після стягування



до стягування

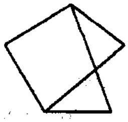


після стягування

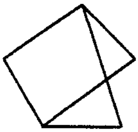


Теорема Зикова. Нехай у графі $\Gamma = [A,B]$ є дві несуміжні вершини u,v. Тоді хроматична функція $\pi(\Gamma,t)$ є сумою двох хроматичних функцій $\pi(\Gamma_1,t)$ *і* $\pi(\Gamma_2,t)$, де граф Γ_1 виходить з Γ з'єднанням несуміжних вершин u,v ребром, а граф Γ_2 виходить з Γ стягуванням вершин u,v.

Ця теорема дозволяє конструктивно описувати хроматичну функцію будь-якого графа. При цьому треба враховувати, що хроматична функція повного графа - це зовсім конкретний об'єкт - багаточлен, що зазначений вище. Приступимо до побудови хроматичної функції в загальній ситуації. Будемо хроматичну функцію $\pi(\Gamma,t)$ графа Γ позначати графічною інтерпретацією графа Γ, тобто картинкою, що зображує вершини і ребра, що їх з'єднують. Таким чином, якщо, наприклад, граф Γ виглядає як:



то ця картинка і буде позначенням функції $\pi(\Gamma,t)$. Якщо потрібно буде записати природне позначення $4\pi(\Gamma,t)$, то будемо застосовувати символ:

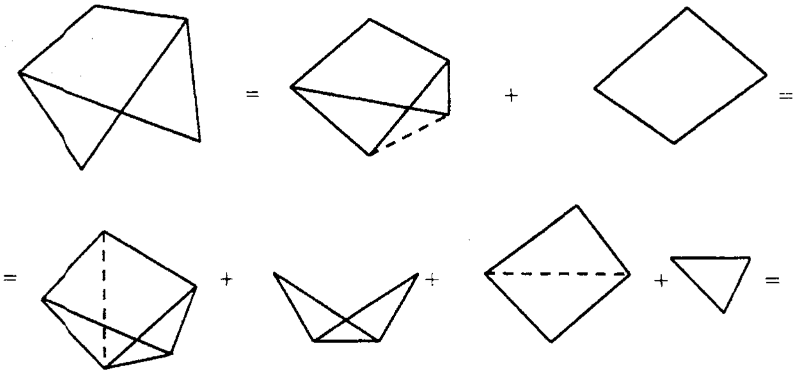
4 

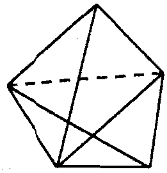
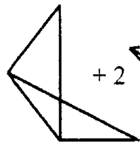
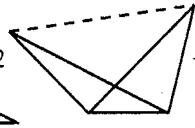
Отже, нехай $\Gamma=[A,B]$ - довільний граф. Якщо він повний, то хроматична функція $\pi(\Gamma,t)$ відома. Якщо граф $\Gamma=[A,B]$ - неповний, то фіксуємо в ньому будь-які дві несуміжні вершини u,v і опишемо $\pi(\Gamma,t)$ відповідно до теореми Зикова:

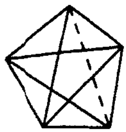
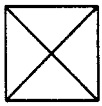
$$\pi(\Gamma,t)=\pi(\Gamma_1,t)+\pi(\Gamma_2,t).$$

До кожного з доданків у правій частині цієї рівності повторно застосуємо теорему Зикова і так далі. Неважко помітити, що ці повторні застосування теореми Зикова стануть неможливими, коли в правій частині виявляться хроматичні функції тільки повних графів. Оскільки останні описані явно, загальна хроматична функція вийде також в явному вигляді. Більш того, зі сказаного випливає, що *будь-яка хроматична функція є багаточленом* (як сума багаточленів). Саме тому хроматичну функцію графа називають його *хроматичним багаточленом*.

Продемонструємо приклад побудови хроматичного багаточлена:



=  +  + 2  + 3  =

=  + 4  + 4  =

$$= t(t-1)(t-2)(t-3)(t-4) + 4t(t-1)(t-2)(t-3) + 4t(t-1)(t-2).$$

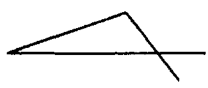
3.19 Ейлерів цикл і ейлерів граф

Задача про Кенігсбергскі мости. Задача полягає в тому, щоб пройти кожний міст по одному разу і повернутися у вихідну точку. План мостів можна звести до графа, у якому ребра відповідають мостам, а вершини - різним розділеним частинам міста.

Ейлер поставив загальну задачу: у яких випадках у скінченному графі можна знайти такий цикл, в якому кожне ребро графа брало б участь один раз? Такий маршрут є простим циклом, який містить усі ребра графа. Такі цикли стали називати ейлеровими, а графи, що мають ейлерові цикли, - ейлеровими графами.

Формальне визначення. Нехай $\Gamma = [A, B]$ - граф. *Ейлеровим циклом* у графі називається такий цикл, що містить усі ребра і усі вершини цього графа. Нагадаємо, що за визначенням у циклах не повторюються ребра. Таким чином, за наявністю ейлерового циклу в графі цей граф можна обійти по всіх ребрах, пройшовши кожне ребро тільки один раз. Граф, що має ейлерів цикл, називається *ейлеровим*. Наприклад:

Нижче наведений приклад графа, що не є ейлеровим:



Теорема. Скінчений неорієнтований граф G є ейлеровим, тоді і тільки тоді, коли він зв'язаний і степені усіх його вершин парні.

Необхідність. Нехай граф G буде ейлеровим. Очевидно, що умова зв`язності тут необхідна. Щоразу, коли ейлерів цикл проходить через якусь вершину, він повинен ввійти в неї по одному ребру і вийти по іншому. Тому умова парності степеня вершин необхідна.

Достатність. Припустимо, що умови зв'язності і парності степеня вершин графа виконуються. Почнемо ланцюг P з довільної вершини a і продовжимо його, поки це можливо, увесь час через нові ребра. Оскільки усі вершини мають парні локальні степені, то ланцюг P повинен закінчитися у вершині a. Якщо ланцюг P містить не всі ребра графа G, то видалимо із графа G частину ланцюга P, що складається з ребер цього циклу.

Оскільки графи P і G мають парні степені вершин, то вершини деякого графа $G\setminus P$ теж повинні бути парні. Оскільки граф G зв'язний, то у доповнені $G\setminus P$ повинна існувати вершина *в*, що належить також і частині P. Починаючи з цієї вершини можна побудувати новий ланцюг P' в $G\setminus P$, який закінчується в вершині *в*. Однак тоді з P і P' можна скласти новий цикл:

$$P_I = P(a,b) \bigcup P' \bigcup P(b,a),$$

який повертається в a і містить більше ребер, ніж P. Якщо P_I не є ейлеровим циклом, то ця побудова повторюється. По закінченні цього процесу, буде побудований ейлерів цикл.

У задачі про Кенігсбергські мости всі вершини мають непарний степінь. Тому її розв'язок не можливий.

Теорема. Нехай G - скінченний зв'язний граф із k вершинами непарного степеня. Тоді мінімальне число ланцюгів, що не повторюються по ребрах, і покривають G, дорівнює $k/2$.

Теорема. Нехай G – скінченний орієнтований зв'язний граф. Для того щоб існував орієнтований цикл, що містить усі ребра по одному разу, необхідна і достатня рівність степенів по вхідних і вихідних ребрах.

Теорема. У скінченному зв'язному графі G завжди можна побудувати орієнтований цикл, що проходить через кожне ребро по одному разу в кожному з двох напрямків.

Для визначення такого циклу можна застосувати таке правило (*Пропозиція Террі*): починаючи з довільної вершини a_0, потрібно йти по деякому ланцюгу P, позначаючи на кожному ребрі напрямок, у якому воно було пройдено. Якщо ми приходимо в вершину g перший раз, то помічаємо перше вхідне ребро. З g завжди виходимо по ребру (g, r), що або взагалі не було пройдено, або ж було пройдено в протилежному напрямку. При цьому по першому вхідному в g ребру можна йти тільки тоді, коли інших можливостей не лишилося. Під час кожного проходження через вершину g у ланцюзі P буде одне вхідне ребро й одне вихідне: отже P може закінчитися тільки в a_0. Можна показати, що P покриває все ребро a по одному разу в кожному напрямку (*Лабораторна робота*).

Задача про лабіринти. Коли Тезей, вбивши Мінотавра, знайшов маршрут по коридорах лабіринту в Кносе, задача про лабіринти стала популярною головоломкою. Лабіринт складається з коридорів і їх перехресть. Їх можна подати у вигляді графів, у яких ребра відповідають коридорам, а вершини відповідають - перехрестям.

У термінах графів задача про лабіринт може бути сформульована таким чином: визначити метод, що дозволяє знайти маршрут у графі, який починається в заданій вершині a_0 і приводить в іншу задану вершину a_1.

Перший систематичний процес для побудови виходу з лабіринту, був запропонований Вінером (1873 рік). Його правила такі: з даної вершини a_0 треба проходити по ребрах графа, обираючи в кожній вершині ще не пройдене ребро. З вершини, від якої далі рухатися не можна, маршрут повертається до такої вершини, де ще невикористані ребра. Остання операція тоді буде складатися в поверненні по всій лінії до вершини a_0. При цьому такий маршрут покриє всі ребра графа. (*Лабораторна робота*).

Розглянемо інший метод. *Метод поступового покривання графа* (*Лабораторна робота*). Нехай ми знаходимося у вершині a_0. До усіх вершин, що знаходяться на відстані 1, підійти легко. Для цього потрібно проходити по різних ребрах у a_0, повертаючи щораз з a_1. Кожне ребро $e=(a_0,a_1)$ позначається один раз, коли ми виходимо по ньому з a_0, в a_1 і воно позначається як відкрите. Якщо виявиться, що в a_1 немає ребер, крім e, то, повернувшись у a_0 позначаємо e як закрите. А якщо деяке інше ребро $e'=(a_0,a_1)$ також увійде з a_0 в a_1, то позначаємо його як закрите з обох кінців; теж саме робиться для будь-якої петлі з a_0.

Щоб потрапити у вершини, що знаходяться від a_0 на відстань 2, беремо деяке відкрите $e=(a_0,a_1)$ ребро і помічаємо його знову. В a_1 відкриті ребра також проходяться і позначаються, причому як закриті, якщо вони ведуть до вже пройдених вершин. Коли все це буде виконано, повертаємося в a_0 із a_1 по раніше позначеному вхідному ребру.

Якщо не залишилося ребер, відкритих в a_1, то це вхідне ребро закривається в a_0. Після повернення в a_0 така сама ж операція здійснюється для інших відкритих ребер, і процес продовжується доти, поки всі ребра в a_0 не будуть позначені двічі. Після того як усі вершини на відстані H пройдені, ми отримаємо таку ситуацію:

- усі відкриті ребра позначені n раз в a_0;
- відкриті ребра в будь-якій вершині a_1 - позначені n-1 раз і т.д.

Щоб підійти до вершини на відстань n+1, ми переходимо послідовно в будь-яку вершину a_1 на відстань 1 і відвідуємо усі вершини на відстань n

від a_1, використовуючи при цьому відкрите ребро і позначаючи їх за описаними правилами. (*Лабораторна робота*).

Розглянемо ще один клас задач, пов'язаних зі схемами доріг. Нехай G - схема доріг з ділянками $(a,\ b)$, що зв'язують різні перехрестя a і $в$. Щоб знайти ланцюг

$$p(a_0,b_0)=(a_0,a_1)...(a_{n-1},a_n),(a_n=b_0)$$

між двома пунктами a_0 і b_0, можна застосувати метод поступового покриття графа. Однак основною задачею про схему доріг є визначення найкоротшого маршруту. Для цього між будь-якими двома пунктами ребру $e=(a,b)$ $(a\neq b)$ приписується вага $\mu(a,b)>0$ і потрібно знайти маршрут, для якого $\mu(p)=\sum_{i=1}^{n-1}\mu(a_i,a_{i+1})$ є мінімальним.

Метод редукції індексу. a_0 позначається числом $\mu(a_0)=0$; кожна з інших вершин позначається числом μ (b) або (∞), що перевершують усі ваги (u,v), для різних ребер (u,v). Повторна редукція здійснюється таким чином: якщо існує така пара вершин u, v із $\mu'(u),\mu^j(v)$ числами, що

$$\mu(u,v)<\mu^j(v)-\mu^i(u),$$

то число $\mu^j(v)$ заміняється на $\mu^{j+1}(v)=\mu^i(u)-\mu(u,v)$.

Крім того, в v вказується, що остання редукція була виконана щодо вершини u. Оскільки граф скінченний, цей процес повинний обриватися. Для вершини a_n ми отримуємо сусідню вершину a_{n-1}, що застосована в останній редукції. Для a_{n-1} є вершина a_{n-2} останньої редукції і т.д. Таким чином отримуємо спадну послідовність індексів.

$$\mu(a_n)=\mu(a_{n-1})+\mu(a_{n-1},a_n)$$
$$\mu(a_{n-1})=\mu(a_{n-2})+\mu(a_{n-2},a_{n-1})$$

....

Очевидно, що відповідний ланцюг P може закінчуватися тільки в a_0. Підсумовуючи, отримуємо

$$\mu(a_n)=\sum_{i=0}^{n-1}\mu(a_i,a_{i+1}).$$

Доведемо, що P є найкоротшим ланцюгом.

Нехай $Q=(a_n,b_1,b_2,....,a_0)$. Інший ланцюг, що проходить через вершину b_i, оскільки ніяка редукція неможлива, повинен бути

$$\mu(a_n) \le \mu(b_1) + \mu(a_n, b_1)$$
$$\mu(b_1) \le \mu(b_2) + \mu(b_1, b_2),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

а сума дає $\mu(a_n) \le \sum \mu(b_i', b_{i+1})$, тобто Q не може бути коротше P.

Розглянемо ще один метод визначення найкоротших маршрутів у схемі доріг (*Лабораторна робота*).

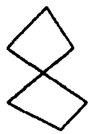
Починаючи з a_0 будемо проходити до усіх вершин, досяжних по одному ребру, і в кожній вершині записувати її відстань від a_0. Потім, як для методу поступового покривання, переходимо до усіх вершин, що можна з'єднати з вершиною a_0 двореберними ланцюгами, і знову помічаємо їх відстань від a_0. Якщо є декілька ланцюгів до такої вершини, то враховуємо тільки найкоротші з них, і вказуємо, від яких вершин вони надходять. Це поступове покривання G від a_0 продовжується доти, поки деякий ланцюг P_1 довжини μ_1 не досягне вершини b_0. Потім із розгляду виключаються усі вершини, відстані яких перевищують μ_1, і побудова продовжується на вершинах, що залишилися. Якщо до b_0 доходимо деяким іншим ланцюгом P_2 довжини $\mu_2 < \mu_1$, то усі вершини з відстанями, більшими μ_2, відкидаються і т.д. Очевидно, що цей процес призводить до знаходження розв'язку задачі.

Задача про розбиття графа на мінімальне число ланцюгів. У зв'язку з ейлеровими графами є одна класична задача *про мінімальне ланцюгове розбиття*. Вона формулюється таким чином. Нехай $\Gamma = [A, B]$ - зв'язний граф і $a_1, \ldots, a_{2k}$ - усі його вершини непарного степеня (нагадаємо, що в кожному графі число вершин непарного степеня парне). Тоді, якщо $k \ne 0$, то мінімальне число ланцюгів графа, що містять у сукупності всі його ребра, дорівнює k, а якщо $k = 0$, то зазначене мінімальне число дорівнює 1. В останній ситуації очевидно, що мова йде про ейлеровий граф. У першій же ситуації є сенс зупиниться на принциповій схемі міркувань, а саме: побудуємо новий граф $\Gamma' = [A', B']$, де $A' = A \cup \{z\}$, тобто додається одна нова вершина і $B' = B \cup \{(z, a_1), (z, a_2), \ldots, (z, a_{2k})\}$, тобто додається ще $2k$ ребер.

Відзначимо, що отриманий граф $\Gamma' = [A', B']$ є зв'язним і всі його локальні степені вже парні, і тому в ньому існує деякий ейлерів цикл C. Здійснимо обхід по циклу C, почавши і закінчивши цей обхід у вершині z. Ті k ланцюгів, про які мова йде в обговорюваному твердженні, будуть виділятися в процесі цього обходу таким чином: обхід починається у вершині z, отже наступною за нею буде деяка вершина з Γ, при цьому вона буде початком першого ланцюга. Після деякого маршруту треба буде знову повернутися у z. Перший ланцюг закінчиться в тій вершині з Γ, з якої здійсниться повернення в z. Потім за тією ж схемою виникає другий ланцюг і т.д.

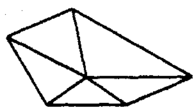
3.20 Гамільтонові цикл і граф. Умови Дірака, Оре і Поша

Гамільтонові цикл і граф. Нехай $\Gamma=[A,B]$ - деякий граф. Він називається *гамільтоновим*, якщо в ньому існує простий цикл, що містить усі вершини графа. Наприклад, кожен повний граф є гамільтоновим, оскільки в ньому проведені всілякі ребра і, зокрема, ті, завдяки яким можливий обхід по усіх вершинах. Нижче наведено приклад графа, що не є гамільтоновим:

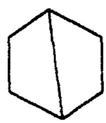


Загальних і легко здійснюваних дій, за допомогою яких можна було б вірогідно з'ясувати, чи є даний граф гамільтоновим, не існує. Однак, існують достатні умови на гамільтоновість, що легко перевіряються. Недоліком тут є те, що навіть якщо жодна з цих умов не виконується, граф може виявитися гамільтоновим. Надалі за замовчуванням будемо вважати, що число вершин у будь-якому графі не менше трьох.

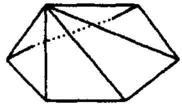
Умова Дірака. Нехай p - число вершин у даному графі. Якщо степінь кожної вершини не менше $p/2$, то граф називається *графом Дірака*. Можна довести, що кожен граф Дірака є обов'язково гамільтоновим. Наприклад:



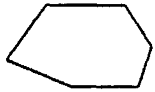
Очевидно, що цей граф є гамільтоновим. Нижче наводиться приклад гамільтонового графа, що не є графом Дірака:



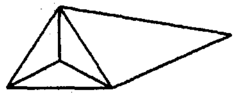
Умова Оре. Як і раніше будемо позначати через p кількість вершин у даному графі. Якщо для будь-якої пари несуміжних вершин x,y виконується нерівність $d(x)+d(y)\geq p$, то граф називається *графом Оре* (словами: степені будь-яких двох несуміжних вершин не менші загального числа вершин у графі). Можна довести, що будь-який граф Оре є обов'язково гамільтоновим. Наприклад:



Нижче наведений приклад графа, що не є графом Оре, проте є гамільтоновим графом:



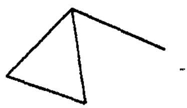
Неважко помітити, що будь-який граф Дірака автоматично є графом Оре. Нижче наведений приклад графа Оре, що не є графом Дірака:



Умова Поша. Це - більш складна конструкція. Введемо таку функцію $f(x)$ цілого додатного аргументу x. Спочатку запишемо визначення формулою, а потім прокоментуємо його. Отже, мова йде як і раніше про граф $\Gamma = [A, B]$, для якого і будується функція $f(x)$:

$$f(x) = \left|\{a \in A | d(a) \le x\}\right|.$$

Написане означає, що функція $f(x)$ для кожного цілого додатного x приймає значення, що дорівнює числу вершин графа $\Gamma = [A, B]$, степінь яких не перевершує x. Таку функцію $f(x)$ називають *функцією Поша* графа $\Gamma = [A, B]$. Для прикладу побудуємо функцію Поша такого графа:



Функція буде описана таблично:

X	0	1	2	3	4	5	...
f(x)	0	1	3	4	4	4	...

Тепер сформулюємо умову Поша. *Графом Поша* називається граф $\Gamma = [A.B]$, що задовольняє такі умови (число вершин цього графа позначимо через p, функцію Поша позначимо через $f(x)$, де символ x буде відповідати цілому числу:

1) для $1 \le x < \frac{p-1}{2}$ виконується нерівність $f(x) < x$;

2) якщо $\frac{p-1}{2}$ - ціле число, то для $x = \frac{p-1}{2}$ має місце нерівність:

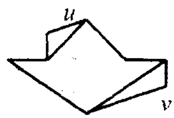
$$f(x) \le \frac{p-1}{2}.$$

Можна довести, що *кожен граф Поша обов'язково є гамільтоновим.* Легко помітити, що простий цикл на великому числі вершин графом Поша не є, однак, зазвичай, є гамільтоновим графом. Крім того, неважко помітити, що будь-який *граф Дірака є графом Поша.* Те ж саме вірно і стосовно графів Оре: *кожен граф Оре є графом Поша.* Зворотне в обох останніх випадках невірне.

Наприклад, фіксуємо будь-який повний граф на досить великому числі вершин. Додамо до нього ще одну вершину і з'єднаємо її з будь-якими двома вершинами вихідного графа. Очевидно, що знову отриманий граф є гамільтоновим. Фіксуємо у ньому будь-який гамільтонів цикл і видалимо з графа будь-яке ребро, не включене в цей цикл. Отриманий таким чином граф буде, як неважко перевірити, графом Поша. Однак, він не буде графом Оре, оскільки будуть відразу дві пари несуміжних вершин, сума степенів яких менша числа вершин у графі.

3.21 Теореми Менгера, Холла і Кьоніга

Формулювання теореми Менгера (вершинний і реберний варіанти). Нехай $\Gamma = [A, B]$ - деякий граф і $u, v \in A$, причому $(u, v) \notin B$, тобто вершини u, v не є суміжними. Набір вершин $x_1, \ldots, x_n \in A$ називається (u, v)-*відокремлювальним*, якщо після видалення з графа усіх вершин набору, вершини u, v виявляються незв'язними, тобто опиняться в різних зв'язних компонентах. Якщо вихідний граф зв'язний, то для кожної пари несуміжних вершин u, v виникає така характеристика: *мінімальне число* (u, v)*-відокремлювальних вершин.* В подальшому будемо позначати це число символом k. Наприклад, у наведеному нижче графі це число дорівнює двом, хоча (u, v)-*відокремлювальних* наборів тут декілька:



Надалі, будь-який ланцюг, що з'єднує вершини u, v, будемо називати (u, v)*-ланцюгом*. Якщо в двох (u, v) - ланцюгах є загальна вершина, відмінна від u, v, то будемо називати такі два (u, v) - ланцюги *вершинно-перетинанними.* Очевидно, що в кожному графі для даних двох вершин u, v існує така характеристика - *максимальне число вершинно-неперетинанних (попарно)* (u, v)*-ланцюгів.* Далі будемо позначати тільки що введену характеристику через l. Неважко зрозуміти, що *завжди* $l \le k$.

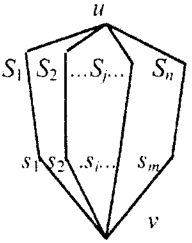
Сформулюємо тепер *теорему Менгера*: *завжди* $l = k$, або те ж саме словами: *мінімальне число* (u, v)*-відокремлювальних вершин дорівнює максимальному числу вершинно-неперетинанних (попарно)* (u, v)*-ланцюгів.*

Сформульована таким чином теорема називається *вершинним варіантом теореми Менгера.* Існує аналогічний реберний варіант цієї теореми. Сформулюємо його. У тих самих позначеннях набір ребер $b_1,...,b_r \in B$ називається (u,v)-*відокремлювальним*, якщо після видалення цих ребер із графа вершини u,v є незв'язаними, тобто в різних зв'язних компонентах. Ясно, що граф має таку характеристику: *мінімальне число ребер, що складають* (u,v)-*відокремлювальний набір.* Позначимо цю характеристику через k.

Назвемо, тепер, набір (u,v)-ланцюгів *реберно-неперетинанним*, а ланцюги цього набору – *реберно-неперетинанними*, якщо будь-які два ланцюги набору не мають загальних ребер. Очевидно, для кожного графа і фіксованих у ньому вершин u,v існує така характеристика: *максимальне число реберно-неперетинанних* (u,v)-*ланцюгів.* Позначимо цю характеритику через l. Як і в попередньому випадку, мабуть, що $l \le k$. *Реберний варіант теореми Менгера: завжди* $l = k$. Або те ж саме словами: *мінімальне число* (u,v)-*відокремлювальних ребер дорівнює максимальному числу реберно-неперетинанних* (u,v)-ланцюгів.

Продемонструємо на двох прикладах корисність теореми Менгера в обох варіантах – вершинному і реберному.

Приклад 1. *Теорема Холла про системи різних представників.* Нехай $S_1,S_2,...,S_n$ - деякі множини і є елементи $s_i \in S_i$, $i=1,2,...,n$. Елементи $s_1,s_2,...,s_n$ називаються *системою різних представників*, якщо усі вони попарно різні. Якщо розглянути множини $S_1=\{1\}, S_2=\{2\}, S_3=\{1,2\}$, то стане ясно, що система різних представників існує не завжди. Теорема Холла стверджує: *система множин* $S_1,S_2,...,S_n$ *має систему різних представників тоді і тільки тоді, коли в об'єднанні будь-яких* k множин *з числа даних* $S_1,S_2,...,S_n$ *є* k *різних елементів.* $k=1,2,...,n$. Для доведення цього факту будується такий граф:



Тут $s_1,...,s_m$ - елементи, що складають об'єднання $\bigcup_{i=1}^{n} S_i$, а ребро (s_i,S_j) включається тоді і тільки тоді, коли $s_i \in S_j$. Саме в цьому графі факт того, що мінімальне число (u,v)-відокремлювальних вершин дорівнює максимальному числу вершинно-неперетинанних (u,v)-ланцюгів, можна інтерпретувати як те, що система різних представників існує тоді і тільки тоді, коли в об'єднані будь-яких k множин міститься не менш k різних елементів.

Приклад 2. *Теорема Кьоніга про покривні лінії.* Нехай є деяка матриця $M=(m_{ij})$, у якій усі елементи рівні або нулю, або одиниці. Словом *лінія* будемо позначати або рядок, або стовпець у даній матриці. Дві одиниці в матриці будемо називати *незалежними*, якщо вони не знаходяться на одній лінії. Довільний набір одиниць будемо називати *набором незалежних одиниць*, якщо будь-які дві одиниці в наборі незалежні. Нарешті, набір ліній даної матриці будемо називати *покривним*, якщо кожна одиниця матриці належить хоча б одній лінії набору.

Теорема Кьоніга стверджує: *максимальне число незалежних одиниць матриці дорівнює мінімальному числу ліній, що складають покривний набір.* Для доведення цієї теореми будується такий самий граф, як і для теореми Хола з тією лише різницею, що $S_1,...,S_n$ стають іменами рядків даної матриці, $s_1,...,s_m$ - стають іменами стовпців і ребро (S_i, s_j) включається тоді і тільки тоді, коли на перетинанні *i-ого* рядка і *j*-ого стовпця в матриці стоїть одиниця. Факт, сформульований у теоремі Кьоніга, саме в цій матриці перетворюється у вершинний варіант теореми Менгера стосовно вершин u,v.

3.22 Паросполучення у графах

Існує багато задач, що пов'язані з давно відомою так званою задачею про весілля. Вона формулюється таким чином. Є скінченна множина хлопців, у кожного з яких є декілька подруг. За яких умов можна оженити хлопців, причому так, щоб кожний оженився на одній із своїх подруг? Тут мається на увазі, що жодна дівчина не виходить заміж більше ніж за одного хлопця.

Нехай $\Gamma=[A,B]$ - граф і $S\subseteq B$ - деяка множина ребер у ньому. Множина S називається *паросполученням*, якщо будь-які його два ребра не мають загальної вершини. Домовимося, що множина з одного ребра теж буде називатися *паросполученням*, як і будь-яка порожня множина.

Паросполучення називається *максимальним*, якщо до нього не можна додати жодного ребра так, щоб знову вийшло паросполучення. Нарешті, паросполучення називається *найбільшим*, якщо воно складається з найбільш можливої кількості ребер. Очевидно, що кожне найбільше паросполучення є максимальним; зворотне невірно, наприклад:

тут товсті ребра - максимальне, але не найбільше паросполучення, а тонкі ребра - найбільше паросполучення. Пошук найбільшого паросполучення в графі є класичною алгоритмічною задачею. Ми розглянемо її розв'язання не для загального випадку, а для *дводольних* графів.

Граф $\Gamma = [A, B]$ називається *дводольним*, якщо його множину вершин A можна описати як об'єднання двох не порожніх підмножин A_1, A_2 без загальних елементів так, що будь-яке ребро з B буде мати один кінець у A_1, а інший кінець - в A_2. Таким чином, немає жодного ребра, що з'єднувало б між собою вершини підмножини A_1 або підмножини A_2.

Якщо вважати, що $A_1 = \{x_1, \ldots, x_r\}, A_2 = \{y_1, \ldots y_s\}$, то дводольний граф можна описати не тільки вже відомою матрицею суміжностей, але й такою *матрицею дводольного графа*: це матриця – розмірністю $r \times s$ і, якщо позначати її загальний елемент через m_{ij}, то

$$m_{ij} = \begin{cases} 1, & \text{якщо } (x_i, y_j) \in B \\ 0, & \text{якщо } (x_i, y_j) \notin B \end{cases}.$$

Очевидно, що така матриця описує граф однозначно, хоча є набагато меншою за обсягом, ніж матриця суміжностей.

Коли множина A_1 складається з n вершин, а множина A_2 складається з m вершин і в дводольному графі проведені *усі* можливі ребра, то говорять, що дводольний граф $\Gamma = [A, B]$ є *повним дводольним графом* і позначають його символом $K_{m,n}$.

Опишемо алгоритм вибору найбільшого паросполучення у дводольному графі на основі введеного поняття матриці дводольного графа.

Крок 0. По матриці $M = (m_{ij}), \quad i = 1, \ldots, p, \quad j = 1, 2, \ldots, q$ дводольного графа будуємо робочу таблицю. Нею є матриця тієї ж розмірності. У клітинку з номером (ij) помістимо символ « $\times$ » і назвемо її *неприпустимою*, якщо в матриці дводольного графа $m_{ij} = 0$, а якщо ж $m_{ij} = 1$, то в клітинку робочої таблиці не запишемо нічого і назвемо таку клітину *припустимою*. Ліворуч від робочої таблиці розмістимо для зручності номери її рядків, а зверху над таблицею розмістимо номери її стовпців. *Крок 1.* Переглянемо зліва направо перший рядок і в першу ж припустиму клітинку першого рядка поставимо символ «1». Якщо в першому рядку всі клітинки неприпустимі, то перейдемо до другого рядка й у першу ж (при перегляді зліва направо) припустиму клітинку поставимо «1». Якщо ж немає припустимих клітинок і в другому рядку, то проставимо зазначеним вище способом «1» у третьому рядку.

Якщо виявиться, що у таблиці всі клітинки неприпустимі, то на цьому всі дії закінчуються і видається відповідь: «у графі немає ребер». Якщо ж все-таки вдасться поставити першу «1», то після цього переглянемо всі інші рядки таблиці в порядку зростання їх номерів. У кожному черговому рядку переглядаємо її клітинки зліва направо і фіксуємо першу по ходу перегляду допустиму клітинку таку, що є незалежною стосовно припустимих клітинок, у яких уже міститься символ «1», і проставляємо в цю клітинку «1».

Після цього переходимо до таких же дій у наступному рядку. Якщо в рядку такої клітинки не виявиться, то переходимо до наступного рядка і виконуємо в ній таку ж перевірку. Фіксуємо набір ребер у графі, що відповідають проставленим у таблиці символам «1». Під ребром, що *відповідає* символу «1» у робочій таблиці, мається на увазі таке: якщо «1» міститься в клітинці з номером (ij), то *відповідним* буде ребро (x_i, y_j). Легко помітити, що цей набір ребер є максимальним паросполученням. Якщо в результаті проведення всіх дій на цьому кроці в кожному рядку робочої таблиці виявиться символ «1», то ребра із зазначеного тільки що паросполучення якраз і складають найбільше паросполучення, і дії закінчені. Якщо ж в результаті першого кроку залишилися рядки без «1», то позначимо їх праворуч від таблиці символом «- » і переходимо до наступного кроку.

Крок 2. Переглянемо всі позначені рядки в порядку зростання їхніх номерів. Перегляд чергового рядка полягає в такому: у рядку відшукуються припустимі клітинки, і стовпці, в яких ці клітинки розташовані, позначаються номером розглядуваного рядка. При цьому дотримується принцип ($\wp$): *якщо позначка уже міститься, то на її місце друга не ставиться.* Позначки, про які тільки що було сказано, фізично виставляються внизу, під таблицею. Якщо у результаті цього кроку жоден зі стовпців не буде позначений, то це означає, що вже наявне паросполучення (отримане на кроці 1) є шуканим найбільшим і всі дії припиняються. Якщо ж серед стовпців з'являться позначені, то переходимо до наступного кроку.

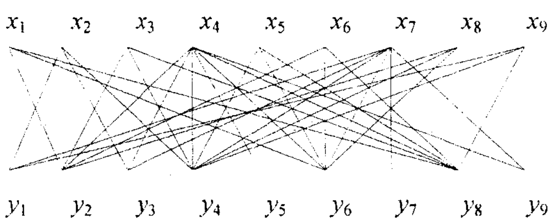
Крок 3. Переглянемо позначені стовпці в порядку зростання їхніх номерів. Перегляд стовпця полягає в такому: у стовпці відшукується символ «1» і рядок, у якому він розташований, помічається номером стовпця, що переглядається. Фізично позначки виставляються праворуч від таблиці, на рівні відповідних рядків. Завжди витримується принцип ($\wp$). Якщо у результаті дій на цьому кроці виникне ситуація, коли в стовпці, що переглядається, немає символу «1», то дії на даному кроці перериваються і треба перейти до наступного кроку. Якщо ж у результаті дій по цьому кроці будуть переглянуті всі позначені раніше стовпці і виникне набір позначених рядків (одні будуть позначені символом « - », інші – номерами стовпців), то треба повернутися до *Кроку 2*. Якщо в результаті цих дій на *Кроці 2* не виникне нових позначених стовпців, то це означає, що наявне паросполучення є шуканим і процес зупиняється. Якщо ж серед стовпців, що позначаються, виникнуть нові позначені стовпці, то слід повторити *Крок 3* з новим набором позначених стовпців. Якщо у результаті цих дій не виникне нових позначених рядків, то це означає, що наявне паросполучення є шуканим.

Крок 4. Розглядаємо стовпець, що має позначку і не містить символ «1». Змінимо набір символів «1», розташованих у робочій таблиці. У даному стовпці поставимо «1» у рядок, номер якого відповідає позначці цього стовпця. Потім у цьому рядку відшукаємо «попередній » символ «1» і перемістимо його у відповідному стовпці у рядок, номер якого дорівнює позначці цього стовпця. Можна довести, що описана процедура завжди здійсненна.

Потім у рядку, куди потрапив останній символ «1» відшукаємо «попередній» символ «1» і з ним проробимо те ж саме. Зрештою черговий переміщуваний символ «1» виявиться у рядку, де більше символів «1» немає. Виникає новий набір символів «1», у якому вже елементів на один більше, ніж у вихідному наборі. За цим новим набором можна побудувати паросполучення так само, як це робилося на самому початку і після цього повторити все знову. Зрештою виникне одне із зазначених вище умов припинення дій за алгоритмом і найбільше паросполучення буде знайдене.

Для прикладу знайдемо найбільше паросполучення у такому дводольному графі:

	1	2	3	4	5	6	7	8	9
1	0	1	0	0	0	1	0	1	0
2	1	0	1	1	0	0	0	0	0
3	0	1	0	1	0	0	0	1	0
4	0	1	1	1	1	1	0	1	1
5	0	0	0	1	0	1	0	1	0
6	0	1	0	0	0	1	0	1	0
7	1	0	1	1	0	1	1	0	1
8	0	1	0	1	0	1	0	0	0
9	1	0	0	1	0	0	0	1	0



Проходимо крок за кроком описаний вище алгоритм. Робоча таблиця має такий вигляд:

	1	2	3	4	5	6	7	8	9	
1	×		×	×	×		×		×	
2		×			×	×	×	×	×	
3	×		×		×	×	×		×	
4	×						×			
5	×	×	×		×		×		×	
6	×		×	×	×		×		×	
7		×			×			×		
8	×		×		×		×	×	×	
9		×	×		×	×	×		×	

Перший крок: проставляємо незалежні одиниці й відмічаємо рядки, в які одиниці не потрапили:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	
2	**1**	×			×	×	×	×	×	
3	×		×	**1**	×	×	×		×	
4	×		**1**				×			
5	×	×	×		×	**1**	×		×	
6	×		×	×	×		×	**1**	×	
7		×			×		**1**	×		
8	×		×		×		×	×	×	
9		×	×		×	×	×		×	

Паросполучення, що відповідає обраним одиницям:

$$(x_1, y_2), (x_2, y_1), (x_3, y_4), (x_4, y_3), (x_5, y_6), (x_6, y_8), (x_7, y_7).$$

Далі стовпці припустимих клітинок позначених рядків позначимо номерами цих рядків, дотримуючись принципу ($\wp$):

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	
2	**1**	×			×	×	×	×	×	
3	×		×	**1**	×	×	×		×	
4	×		**1**				×			
5	×	×	×		×	**1**	×		×	
6	×		×	×	×		×	**1**	×	
7		×			×		**1**	×		
8	×		×		×		×	×	×	
9		×	×		×	×	×		×	
	9	8		8		8		9		

Далі у позначених стовпцях відшукаємо одиниці і їхні рядки позначимо номерами їхніх стовпців:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	2
2	**1**	×			×	×	×	×	×	1
3	×		×	**1**	×	×	×		×	4
4	×		**1**				×			
5	×	×	×		×	**1**	×		×	6
6	×		×	×	×		×	**1**	×	8
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9		×	×		×	×	×		×	-
	9	8		8		8		9		

Тепер знову позначимо стовпці, відправляючись від позначених рядків і дотримуючись принципу ($\wp$):

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	2
2	**1**	×			×	×	×	×	×	1
3	×		×	**1**	×	×	×		×	4
4	×		**1**				×			
5	×	×	×		×	**1**	×		×	6
6	×		×	×	×		×	**1**	×	8
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9		×	×		×	×	×		×	-
	9	8	2	8		8		9		

Тепер знову переглянемо позначені стовпці і позначимо рядки:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	2
2	**1**	×			×	×	×	×	×	1
3	×		×	**1**	×	×	×		×	4
4	×		**1**				×			3
5	×	×	×		×	**1**	×		×	6
6	×		×	×	×		×	**1**	×	8
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9		×	×		×	×	×		×	-
	9	8	2	8		8		9		

Тепер знову позначимо стовпці (за принципом ($\wp$)):

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2	1	×			×	×	×	×	×	1
3	×		×	1	×	×	×		×	4
4	×		1				×			3
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	8
7		×			×		1	×		
8	×		×		×		×	×	×	[illegible]
9		×	×		×	×	×		×	-
	9	8	2	8	4	8		9	4	

Тепер - знову рядки:

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2	1	×			×	×	×	×	×	1
3	×		×	1	×	×	×		×	4
4	×		1				×			3
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	8
7		×			×		1	×		7
8	×		×		×		×	×	×	-
9		×	×		×	×	×		×	-
	9	8	2	8	4	8		9	4	

У результаті склалася ситуація, коли в стовпці №9 з позначкою «4» немає одиниці. Отже, можна набір одиниць збільшити (нові одиниці позначимо курсивом):

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2	1	×	*1*		×	×	×	×	×	1
3	×		×	1	×	×	×		×	4
4	×		1				×		*1*	3
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	8
7		×			×		1	×		7
8	×		×		×		×	×	×	-
9	*1*	×	×		×	×	×		×	-
	9	8	2	8	4	8		9	4	-

Ми отримали новий набір незалежних одиниць:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	
2		×	**1**		×	×	×	×	×	
3	×		×	**1**	×	×	×		×	
4	×						×		**1**	
5	×	×	×		×	**1**	×		×	
6	×		×	×	×		×	**1**	×	
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9	**1**	×	×		×	×	×		×	

Тепер уся процедура повторюється спочатку, і на останній таблиці вже зазначена єдина позначка «-» у рядку №8. Помітимо стовпці припустимих клітинок цього рядка її номером:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	
2		×	**1**		×	×	×	×	×	
3	×		×	**1**	×	×	×		×	
4	×						×		**1**	
5	×	×	×		×	**1**	×		×	
6	×		×	×	×		×	**1**	×	
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9	**1**	×	×		×	×	×		×	
		8		**8**		**8**				

Потім в позначених стовпцях знайдемо одиниці і їхні рядки позначимо номерами стовпців:

	1	2	3	4	5	6	7	8	9	
1	×	**1**	×	×	×		×		×	**2**
2		×	**1**		×	×	×	×	×	
3	×		×	**1**	×	×	×		×	**4**
4	×						×		**1**	
5	×	×	×		×	**1**	×		×	**6**
6	×		×	×	×		×	**1**	×	
7		×			×		**1**	×		
8	×		×		×		×	×	×	-
9	**1**	×	×		×	×	×		×	
		8		**8**		**8**				

Потім стовпці припустимих клітинок позначених рядків:

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2		×	1		×	×	×	×	×	
3	×		×	1	×	×	×		×	4
4	×						×		1	
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	
7		×			×		1	×		
8	×		×		×		×	×	×	
9	1	×	×		×	×	×		×	
		8		8		8		1		

Потім знову рядки:

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2		×	1		×	×	×	×	×	
3	×		×	1	×	×	×		×	4
4	×						×		1	
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	8
7		×			×		1	×		
8	×		×		×		×	×	×	
9	1	×	×		×	×	×		×	
		8		8		8		1		

Потім знову стовпці:

	1	2	3	4	5	6	7	8	9	
1	×	1	×	×	×		×		×	2
2		×	1		×	×	×	×	×	
3	×		×	1	×	×	×		×	4
4	×						×		1	
5	×	×	×		×	1	×		×	6
6	×		×	×	×		×	1	×	8
7		×			×		1	×		
8	×		×		×		×	×	×	
9	1	×	×		×	×	×		×	
		8		8		8		1		

Склалася ситуація, коли розміщення позначок «зациклилося». Це означає, що шукане паросполучення складається з ребер, що відповідають проставленим одиницям.

Задача про призначення на вузькі місця. Існує n робочих місць $p_1,...,p_n$ на деякому конвеєрі і n робітників $w_1,...,w_n$, яких потрібно на ці робочі місця розставити. Відома продуктивність C_{ij} робітника w_i на робочому місці p_j. Той факт, що для деякого розподілу на робочі місця робітник w_k потрапляє на робоче місце p_{i_k} можна описати такою таблицею:

$$\sigma = \begin{pmatrix} 1 & 2 & ... & k & ... & n \\ i_1 & i_2 & ... & i_k & ... & i_n \end{pmatrix}.$$

Маючи матрицю σ, можна знайти для неї мінімальну продуктивність C_{ki_k} і відзначити, що саме ця мінімальна продуктивність і визначає швидкість і продуктивність конвеєра. Те робоче місце, на якому реалізується мінімальна продуктивність і називають *вузьким місцем у призначенні.*

Наведемо алгоритм розв'язання поставленої задачі .

Крок 0. Фіксуємо матрицю продуктивності $C = (C_{ij})$ і будь-яке призначення на робочі місця (наприклад, робітник w_i призначається на робоче місце p_i, $i = 1,...,n$). Нехай s - мінімальна продуктивність для цьому призначення. Побудуємо робочу таблицю тієї ж розмірності, що і матриця C. У клітинку з номером (ij) в цій таблиці проставимо символ «×», якщо $C_{ij} \le s$. Інакше цю клітинку залишимо порожньою.

Крок 1. Розглядаючи робочу таблицю, побудовану на попередньому кроці, як робочу таблицю в алгоритмі для вибору найбільшого паросполучення в дводольному графі, знайдемо відповідне найбільше паросполучення. Якщо у ньому виявиться n ребер, то по них відновлюється нове призначення на робочі місця і з новою, вищою ніж попередня, мінімальною продуктивністю. Позначимо її знову через s і повернемося до *Кроку 0.* Якщо ж число ребер виявиться менше n, то наявне призначення на робочі місця вже є оптимальним.

3.23 Мережі і потоки у мережах. Алгоритм Форда-Фалкерсона

Нехай $\overline{\Gamma} = [A, B]$ - деякий орграф і $f : B \to \Re$ - функція на множині ребер. Тоді пара $\overline{\Gamma}, f$ називається *мережею*, а функція f в контексті мережі називається *функцією пропускної здатності* або *пропускною здатністю* мережі.

Будь-яка функція $g : B \to \Re$, що задовольняє нерівність $g \le f$, називається *потоком у мережі.* В обговоренні властивостей потоків у мережі традиційно використовується таке позначення. Нехай $h : B \to \Re$ - будь-яка функція і $U, V \subseteq A$ - дві будь-яких підмножини вершин. $h(U,V)$ означає суму значень функції h на ребрах $(x, y) \in B$ таких, що $x \in U$ і $y \in V$. Якщо

U містить єдину вершину a, то $h(a,V)$ означає суму ваг ребер, що починаються в U і закінчуються у вершинах з V. Аналогічно $h(U,a)$ є сумою значень функції h на ребрах, що починаються в U і закінчуються в a. Потік $g: B \to \Re$ у мережі $\overline{\Gamma}, f$ називається *стаціонарним*, якщо існують дві вершини $u, v \in A$ і число $w \in \Re$ такі, що виконуються такі умови:

1) $g(u, A) - g(A, u) = w$;
2) $g(v, A) - g(A, v) = -w$;
3) $g(x, A) - g(A, x) = 0 \quad$ *для* $\quad \forall x \in A, x \neq u, x \neq v$.

У цій ситуації число w називається *величиною* потоку g, вершина u називається *джерелом*, а вершина v - *стоком* потоку g.

Відома така класична *задача про максимальний потік*: у даній мережі для даного джерела і для даного стоку знайти стаціонарний потік максимально можливої величини. Можна довести, що така задача завжди має розв'язок. Один із способів її розв'язання називається *алгоритмом Форда-Фалкерсона*. Сформулюємо цей алгоритм по кроках.

Крок 0. Фіксуємо на мережі $\overline{\Gamma}, f$ з джерелом u і стоком v довільний стаціонарний потік g, наприклад - потік, тотожно дорівнює нулю (тобто дорівнює нулю на кожному ребрі даного орграфа $\overline{\Gamma} = [A, B]$). Неважко перевірити, що такий потік дійсно стаціонарний і має величину 0.

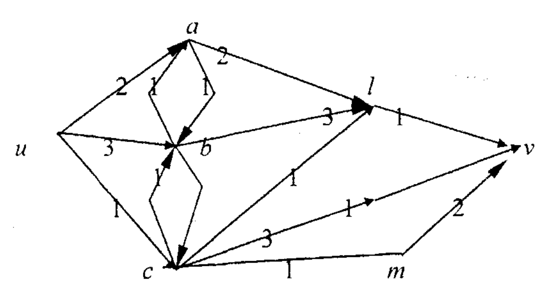
Крок 1. Біля вершини u поставимо позначку такого вигляду: $(-, \infty)$. Тут символ ∞ позначає число, що свідомо перевершує всі числа, що будуть брати участь у подальшому розгляді (у випадку програмування це - комп'ютерна нескінченність, тобто найбільше число, що допускається даним програмним засобом).

Зауваження. Майже всі подальші дії за алгоритмом є розміщенням позначок біля деяких вершин. Мета цього розміщення в тому, щоб зрештою поставити позначку в v стоці або встановити, що стік v позначити неможливо. У першому випадку виявиться можливим замінити наявний стаціонарний потік g на інший стаціонарний потік, що має величину, більшу, ніж величина потоку g. Після цього треба буде запустити все спочатку. В другому випадку виявиться, що наявний потік g оптимальний, тобто його величина має максимально можливе значення. Кожна позначка, крім уже проставленої біля джерела u, буде мати вигляд $(x^{\pm}, \varepsilon)$, де ε - деяке число, а x - ім'я однієї з вершин орграфа $\overline{\Gamma} = [A, B]$, причому реально в позначці це ім'я x буде або у вигляді x^{+}, або у вигляді x^{-}.

Крок 2. Нехай $(x, y) \in B$ - деяке ребро, початок якого, тобто вершина x, уже має деяку позначку $(z^{\pm}, \varepsilon)$ (або, якщо x - це джерело u, то позначку $(-, \varepsilon)$, де $\varepsilon = \infty$). Якщо $g((x, y)) = f((x, y))$, тобто потік g рівний на ребрі (x,y) пропускній здатності f, то позначка біля вершини y не проставляється. Якщо ж $g((x,y)) < f((x,y))$, то позначка біля вершини y проставляється таким чином. На першому місці в позначці буде стояти символ x^+, тобто позначка буде такою: (x^+,μ), де число μ ще потрібно знайти. Покладемо $\mu = \min\{\varepsilon, f((x,y)) - g((x,y))\}$. Нехай тепер $(x,y) \in B$ таке ребро, у якого позначка має кінець, тобто вершина y має позначку $(z^\pm,\varepsilon)$. Якщо $f((x,y)) \le 0$, то позначку при x не виставляють; якщо ж $f((x,y)) > 0$, то x отримує позначку (y^-,μ), де $\mu = \min\{\varepsilon, f((x,y))\}$. Процедура розміщення позначок відповідно *Кроку 2* здійснюється доти, поки не виявиться позначеною вершина v, або доти, поки не з'ясується, що вершину v позначити неможливо. Можна довести, що в останньому випадку потік g, за результатами *Кроку 2, має максимально можливе значення*, і задача розв'язана. Якщо ж вершина v виявилася позначеною, то переходимо до наступного кроку. Відзначимо принципову подробицю: якщо вершина v виявилася позначеною, то *число, що фігурує в позначці, обов'язково позитивне.*

Крок 3. Нехай вершина v має позначку $(x^\pm,\varepsilon)$. Змінимо потік g на декількох ребрах графа, в результаті чого вийде новий стаціонарний потік з джерела u в стік v, значення якого буде на ε (це число зазначене в позначці стоку v) більше значення потоку g. Якщо вершина v має позначку (x^+,ε), то на ребрі (x,v) змінимо потік g, додавши до його значення на цьому ребрі число ε. Якщо вершина v має позначку (x^-,ε), то на ребрі (v,x) змінимо потік g, віднімаючи від нього число ε. Потім перейдемо до вершини x і проробимо те ж, що робили щодо вершини v. При цьому додавати або віднімати будемо колишнє число ε. Продовжуючи так, ми прийдемо до джерела u, що і буде означати завершення зміни потоку. Можна довести, що отриманий таким чином потік є *стаціонарним* і його величина буде на ε більша величини вихідного потоку g. Потім потрібно повторити усе спочатку з уже новим базовим стаціонарним потоком.

Приклад. Знайти максимальний стаціонарний потік з u у v у мережі, що наведена на рисунку нижче (числа означають пропускну здатність).



Вважаємо, що вихідний стаціонарний потік тотожно рівний нулю. Проставляємо позначку біля вершини u: вона така - $(-,\infty)$. Вибираємо далі для позначки вершину a; відповідна позначка: $(u^+,2)$. Вибираємо для позначки вершину l; відповідна позначка: $(a^+,2)$. Тепер з'явилася можливість позначити і вершину v; відповідна позначка: $(l^+,1)$. Виникла можливість збільшити потік на одиницю. Для цього покладемо ребро (l,v) рівний 1 (а не нулю, як це було для вихідного потоку), також рівний 1 новий потік буде і на ребрах (a,l) і (u,a).

На інших ребрах потік залишається рівним нулю. Новий потік має величину один, він стаціонарний з джерелом u і стоком v. Повторимо тепер процедуру спочатку, прагнучи поставити позначку до стоку v. Вершину u позначимо позначкою $(-,\infty)$. Далі позначимо вершину c; відповідна позначка: $(u^+,1)$. Далі позначимо вершину m; відповідна позначка: $(c^+,1)$. Тепер позначимо вершину v; відповідна позначка: $(m^+,1)$. Тепер знову збільшимо на одиницю значення потоку на ребрах $(m,v),(c,m),(u,c)$. Новий потік буде теж стаціонарний, але зі значенням два. Знову поставимо позначку $(-,\infty)$ у вершини u і спробуємо збільшити наявний стаціонарний потік зі значенням два. Позначимо вершину b; відповідна позначка: $(u^+,3)$. Далі позначимо вершину c; відповідна позначка: $(b^+,1)$. Далі позначимо вершину v; відповідна позначка: $(c^+,1)$. Знову змінимо потік на одиницю: додамо одиницю до значень колишнього потоку на ребрах (c,v), $(b,c),(u,b)$. Отриманий потік має значення три.

Подальші спроби досягти позначками вершину v не мають успіху. Отже, максимальний стаціонарний потік знайдений.

4 ЕЛЕМЕНТИ КОМБІНАТОРИКИ

4.1 Перестановки, розміщення і сполучення. Біном Ньютона

Нехай $M=\{1,2,..,n\}$ - будь-яка скінченна множина.

З'єднанням елементів множини M називається будь-який набір, складений з елементів множини M.

Якщо у цьому наборі будь-який елемент зустрічається більше ніж один раз то говорять про *з'єднання з повтореннями.*

Якщо ж у наборі кожен елемент з'являється лише один раз, то говорять про *з'єднання без повторень.*

Надалі ми будемо говорити саме про з'єднання без повторень і тому будемо позначати такі набори просто терміном *з'єднання.*

Перестановкою елементів множин M називається будь-яке з'єднання елементів множини M, у якому обов'язково є всі елементи з M і в якій враховується порядок слідування елементів один за одним. Наприклад, якщо $n=3$, то $(1,2,3)$ і $(3,2,1)$ є різними перестановками.

Неважко довести, що для довільного n кількість P_n усіляких перестановок множини $M=\{1,2,..,n\}$ дорівнює $n!$ ($n!$ - це традиційне позначення для добутку чисел $1\bullet 2\bullet 3\bullet ...\bullet n$; читається «ен факторіал»).

Будь-яке з'єднання з k елементів множини $M=\{1,2,..,n\}$, у якому враховується порядок проходження елементів один за одним, називається *розміщенням з n по k*. Для $k=n$ - це перестановка, для $k>n$ таких з'єднань немає, а для $k<n$ *неважко отримати формулу для кількості A_n^k розміщень з n по k*:

$$A_n^k = n(n-1)(n-2)...(n-k+1),$$

при цьому очевидно, що $A_n^n = P_n$.

Усяке з'єднання з k елементів множини $M=\{1,2,..,n\}$, де $k\le n$, у якому порядок проходження елементів один за одним не враховується, називається *сполученням з n по k*. Наприклад, для $n=4$ з'єднання $(3,1,4)$ і $(4,3,1)$ є різними розміщеннями з чотирьох по три, але як сполучення вони рівні. *Кількість C_n^k сполучень з n по k визначається такою формулою:*

$$C_n^k = \frac{n(n-1)(n-2)....(n-k+1)}{k!}.$$

Числа C_n^k часто називають *біноміальними коефіцієнтами*, оскільки якщо у виразі $(x+y)^n$ розкрити дужки і привести подібні члени, то буде отримана рівність, що називається *біномом Ньютона*:

$$(x+y)^n = x^n + C_n^1 x^{n-1} y + C_n^2 x^{n-2} y^2 + ... + C_n^k x^{n-k} y^k + ... + C_n^{n-1} x y^{n-1} + y^n .$$

Якщо домовитися, що символ C_n^0 означає число один, то біном Ньютона можна записати коротше за допомогою знака суми

$$(x+y)^n = \sum_{k=0}^{n} C_n^k x^{n-k} y^k .$$

Біноміальні коефіцієнти C_n^k характеризуються численними властивостями, яким приділялася увага математиків різних поколінь. Відзначимо три найпростіших з них.

Властивість 1. $C_n^k = C_n^{n-k}, \quad k=0,1,2,...,n$.

Властивість 2. $C_n^0 + C_n^1 + ... + C_n^k + ... + C_n^n = 2^n$.

Властивість 3. $C_n^0 - C_n^1 + C_n^2 - ... + (-1)^k C_n^k + ... + (-1)^n C_n^n = 0$.

Перша властивість Встановлюється порівнянням формул, а останні дві випливають з бінома Ньютона, де змінним x, y дають значення ± 1.

Приклад. У виразі $(x+2y)^{10}$ розкрили дужки і привели подібні члени. Який коефіцієнт буде стояти біля виразу x^4y^6 ?

Щоб відповісти на це запитання розглянемо біном Ньютона:

$$(x+2y)^{10} = x^{10} + C_{10}^{1}x^{9}(2y) + C_{10}^{2}x^{8}(2y)^{2} + \ldots + C_{10}^{6}x^{4}(2y)^{6} + \ldots + (2y)^{10},$$

звідки шукане число дорівнює

$$C_{10}^{6} \times 2^{6} = 64C_{10}^{4} = 64\frac{10.9.8.7}{1.2.3.4} = 64\frac{210}{1} = 13440.$$

4.2 Метод включення-виключення елементів множин

Існує класичний спосіб описування елементів деякої множини з деякими особливостями, що називається *методом включення-виключення*. Сформулюємо відповідну задачу.

Нехай S - деяка скінченна множина і $P = \{p_1, p_2, \ldots, p_n\}$ - список властивостей, які можуть мати і не мати елементи з S. Потрібно визначити формулу, що описує кількість елементів, які не мають жодної із властивостей заданого списку, через будь-які величини, що обчислюються.

Спосіб, що описується нижче, називається *методом включення-виключення*. Нехай $S(p_{i_1}p_{i_2}\ldots p_{i_k})$ означає кількість елементів множини S, що характеризуються властивостями $p_{i_1}, p_{i_2}, \ldots, p_{i_k}$ з $P = = \{p_1, p_2, \ldots, p_n\}$. Шукану кількість елементів позначимо через N, а кількість елементів у S позначимо через M. Можна довести, що має місце така формула (її так само називають *формулою включення-виключення*):

$$N = M - \sum_{i=1}^{n} S(p_i) + \sum_{i,j} S(p_i p_j) - \ldots + (-1)^k \times$$

$$\times \sum_{p_{i_1}, p_{i_2}, \ldots, p_{i_k}} S(p_{i_1} p_{i_2} \ldots p_{i_k}) + \ldots + (-1)^n S(p_1 p_2 \ldots p_n),$$

де підсумовування здійснюється за будь-якими сполученнями властивостей з множини $P = \{p_1, p_2, \ldots, p_n\}$: у першому випадку - за сполученнями по одній властивості, у другому випадку - за сполученнями по двох властивостях і так далі, у k-ому випадку - за сполученнями по k властивостях. Розберемо її докладніше для двох прикладів. Перший з них називається *задачею про відсутність порядку*. Тут розглядаються усі можливі перестановки на n символах. Як відомо, їх загальна кількість дорівнює $n!$. Кожну перестановку $(i_1, \ldots, i_n)$ символів $(1, 2, \ldots, n)$ зв'яжемо з матрицею

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & k & \dots & n \\ i_1 & i_2 & \dots & i_k & \dots & i_n \end{pmatrix},$$

яка називається *підстановкою*.

Прийнято говорити, що *підстановка* σ переводить елемент 1 у елемент i_1, елемент 2 у елемент i_2, ..., елемент k у елемент i_k,... елемент n у елемент i_n. При цьому пишуть: $\sigma(k) = i_k$, $k = 1,...,n$.

Якщо $\sigma(k) = k$, то говорять, що підстановка σ *залишає елемент k на місці*. Підстановка, у якій на місці не залишається жоден елемент, називаються *відсутністю порядку*. Для $n = 3$, наприклад, неважко перерахувати всі підстановки взагалі і вказати серед них відсутності порядку:

підстановки для $n = 3$: $\begin{pmatrix} 123 \\ 123 \end{pmatrix}, \begin{pmatrix} 123 \\ 213 \end{pmatrix}, \begin{pmatrix} 123 \\ 132 \end{pmatrix}, \begin{pmatrix} 123 \\ 231 \end{pmatrix}, \begin{pmatrix} 123 \\ 312 \end{pmatrix}, \begin{pmatrix} 123 \\ 321 \end{pmatrix}$.

Відсутностями порядку серед них є: $\begin{pmatrix} 123 \\ 312 \end{pmatrix}, \begin{pmatrix} 123 \\ 231 \end{pmatrix}$.

Про те, яка може бути кількість відсутностей порядку у загальному випадку, тобто для довільного n, відповідь можна отримати за допомогою методу включення-виключення. Відповідна задача називається *задачею про відсутність порядку*.

Множину усіх підстановок позначимо через S, а список властивостей $P = \{p_1,...,p_n\}$ буде складатися з властивостей p_i: властивість p_i - це властивість тієї чи іншої підстановки залишати на місці елемент i. Очевидно, що відсутність порядку - це саме така підстановка, у якій немає жодної властивості з $P = \{p_1,...,p_n\}$. Помітимо, що у вищенаведених позначеннях кількість $S(p_{i_1},...,p_{i_k})$ підстановок, що залишають на місці елементи $i_1,...,i_k$, дорівнює $(n-k)!$. Тому, виходячи з формули включення-виключення, маємо:

$$N = n! - C_n^1(n-1)! + C_n^2(n-2)! - ... + (-1)^k C_n^k(n-k)! + ... + (-1)^n =$$
$$= n!\left(1 - 1 + \frac{1}{2!} - \frac{1}{3!} + ... + \frac{(-1)^k}{k!} + ... + \frac{(-1)^n}{n!}\right) \approx \frac{n!}{e}.$$

Остання наближена рівність основана на розкладанні за Тейлором функції e^x у точці $x = -1$. Кількість відсутностей порядку на n символах будемо позначати символом U_n.

Другим прикладом застосування формули включення-виключення є *задача про зустрічі*. Її можна сформулювати таким чином. Є множина підстановок на n символах і задане фіксоване ціле невід'ємне число $k \le n$. Скільки підстановок залишають на місці рівно k елементів?

Відповідну відповідь будемо позначати символом $M_{n,k}$. Очевидно, $M_{n,0}=U_n$, $M_{n,n}=1$. Кожна з цих підстановок називається *зустріччю* (порядку k). Можна показати, що $M_{n,k}=C_n^k U_{n-k}$.

4.3 Формальні степеневі ряди і дії над ними

Нехай $a_0,a_1,...,a_n,...$ - числова послідовність. Позначимо *формальний степеневий ряд* символом виду $f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+...$, у якому числа $a_0,a_1,...,a_n,...$ називаються *коефіцієнтами*, а символ t називається *змінною*. Фрагменти a_nt^n степеневого ряду називаються *доданками*. Будь-який многочлен можна вважати записом формального степеневого ряду, у якому всі коефіцієнти, починаючи з якогось номера, дорівнюють нулю. При цьому замість $f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+...$ часто пишуть $f(t)=\sum_{n=0}^{\infty}a_nt^n$.

Два формальних степеневих ряди $f(t)=\sum_{n=0}^{\infty}a_nt^n$ і $g(t)=\sum_{n=0}^{\infty}b_nt^n$ вважаються *рівними*, якщо для кожного n виконується рівність: $a_n=b_n$.

Над формальними степеневими рядами виконуються деякі дії, що є узагальненням арифметики багаточленів.

Додавання. Нехай

$$f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+... \quad \text{і} \quad g(t)=b_0+b_1t+b_2t^2+...+b_nt^n+...$$

два формальних степеневих ряди від однієї і тієї ж змінної t, то *додаванням* цих рядів називається дія, у результаті якої виникає третій формальний степеневий ряд $h(t)=c_0+c_1t+...+c_nt^n+...$, що називається *сумою рядів* $f(t)$ і $g(t)$, причому такий, що $c_n=a_n+b_n$, $n=0,1,2,...$. Таким чином, $\sum_{n=0}^{\infty}a_nt^n+\sum_{n=0}^{\infty}b_nt^n=\sum_{n=0}^{\infty}(a_n+b_n)t^n$. Неважко перевірити, що для будь-яких трьох формальних степеневих рядів $x(t),y(t),z(t)$ має місце рівність: $(x(t)+y(t))+z(t)=x(t)+(y(t)+z(t))$, а для будь-яких двох формальних степеневих рядів $x(t),y(t)$ вірна рівність: $x(t)+y(t)=y(t)+x(t)$. Із сказаного випливає, що формальний степеневий ряд $O(t)=0$ грає у додаванні роль нуля. Крім того, якщо $f(t)=\sum_{n=0}^{\infty}a_nt^n$ - формальний степеневий ряд, то ряд $g(t)=\sum_{n=0}^{\infty}(-a_n)t^n$ грає роль *протилежного елемента*: $f(t)+g(t)=O(t)$.

Віднімання. Якщо

$$f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+... \text{ і } g(t)=b_0+b_1t+b_2t^2+...+b_nt^n+...$$

два формальних степеневих ряди, то ряд $h(t)=c_0+c_1t+...+c_nt^n+...$, що отримується за правилом $c_n=a_n-b_n$, $n=0,1,2,...$, називається *різницею*, а дія, що утворює різницю, називається *відніманням*. Очевидно, що віднімання - це додавання протилежного ряду.

Множення. Нехай

$$f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+... \text{ і } g(t)=b_0+b_1t+b_2t^2+...+b_nt^n+...$$

два формальних степеневих ряди; побудуємо третій формальний степеневий ряд $h(t)=c_0+c_1t+...+c_nt^n+...$, що буде називатися *добутком* рядів $f(t)$ і $g(t)$, а дія, що приводить до добутку, буде називатися *множенням*:

$$\begin{aligned}
&c_0=a_0b_0;\\
&c_1=a_1b_0+a_0b_1;\\
&c_2=a_2b_0+a_1b_1+a_0b_2;\\
&\dots\dots\dots\dots\dots\dots\dots\dots\\
&c_n=a_nb_0+a_{n-1}b_1+...+a_{n-k}b_k+...+a_0b_n;\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots .
\end{aligned}$$

Для множення зберігається традиційна символіка: добуток $h(t)$ рядів $f(t)$ і $g(t)$ записується у вигляді $h(t)=f(t)g(t)$. Очевидно, що для будь-яких трьох рядів $x(t),y(t),z(t)$ справедливі рівності:

$$\begin{aligned}
&(x(t)y(t))z(t)=x(t)(y(t)z(t));\\
&x(t)(y(t)+z(t))=x(t)y(t)+x(t)z(t);\\
&x(t)y(t)=y(t)x(t).
\end{aligned}$$

Ряд $e(t)=1$ характеризується тією властивістю, що притаманна в числовому множенні одиниці для будь-якого формального степеневого ряду $f(t)$, тобто має місце рівність: $f(t)e(t)=f(t)$. Якщо для деяких двох формальних степеневих рядів $f(t)$ і $g(t)$ виконується рівність $f(t)g(t)=1$, то ці ряди називаються *оберненими*. При цьому пишуть $f(t)=g(t)^{-1}$, $g(t)=f(t)^{-1}$.

Ділення. Це дія, за якою по двох формальних степеневих рядах

$$f(t)=a_0+a_1t+a_2t^2+...+a_nt^n+... \text{ і } g(t)=b_0+b_1t+b_2t^2+...+b_nt^n+...$$

з обов'язковим припущенням, що $b_0 \neq 0$, утворюється такий третій ряд $h(t) = c_0 + c_1 t + ... + c_n t^n + ...$, для якого виконується рівність: $f(t) = g(t)h(t)$.

Розписуючи останню рівність по коефіцієнтах маємо:

$a_0 = b_0 c_0$; отже $c_0 = a_0 b_0^{-1}$;

$a_1 = b_1 c_0 + b_0 c_1$; отже $c_1 = (a_1 - b_1 c_0) b_0^{-1}$;

....

$a_n = b_n c_0 + ... + b_0 c_n$; отже $c_n = (a_n - b_n c_0 - ... - b_1 c_{n-1}) b_0^{-1}$;

....

Ряд $h(t) = c_0 + c_1 t + ... + c_n t^n + ...$ називається *часткою від ділення* ряду $f(t)$ на ряд $g(t)$, при цьому часто пишуть: $h(t) = \dfrac{f(t)}{g(t)}$.

Відзначимо для прикладу, що:

$$\frac{1}{1-t} = 1 + t + t^2 + ... + t^n + ...; \quad \frac{1}{1+t} = 1 - t + t^2 - ... + (-1)^n t^n + ... \ .$$

Формальний степеневий ряд $f(t) = \sum_{n=0}^{\infty} a_n t^n$ називається *породною функцією* послідовності $a_0, a_1, ..., a_n, ...$. При цьому дії над рядами, аналогічні діям, що розглядались вище. Приклади застосування техніки дій над функціями, що породжують послідовність, розглядаються у наступному підрозділі.

4.4 Лінійні рекурентні співвідношення

Нехай $u_0, u_1, ..., u_n, ...$ - числова послідовність з такою властивістю:

$$u_n = a_1 u_{n-1} + a_2 u_{n-2} + ... + a_r u_{n-r} \ , (n \geq r),$$

де r фіксоване, а числа $a_1, a_2, ..., a_r$ вважаються наперед заданими.

Таким чином, дана послідовність цілком визначається своїми першими членами $u_0, u_1, ..., u_{r-1}$. Рівність, завдяки якій це виявляється можливим, називається *лінійним рекурентним співвідношенням*. Звідси маємо таку класичну задачу: яким чином визначити загальний член u_n даної послідовності, причому не через попередні члени послідовності, а у вигляді аналітичного виразу від n?

Наведемо рішення цієї задачі за допомогою породної функції.

Нехай $f(t) = u_0 + u_1 t + u_2 t^2 + ... + u_n t^n + ...$ - породна функція послідовності $u_0, u_1, ..., u_n, ...$, яка задається лінійним рекурентним співвідношенням

$u_n = a_1 u_{n-1} + a_2 u_{n-2} + \ldots + a_r u_{n-r}$. Фіксуємо формальний степеневий ряд:

$$g(t) = 1 - a_1 t - a_2 t^2 - \ldots - a_r t^r,$$

де усі коефіцієнти, починаючи з $(r+1)$-го степеня, дорівнюють нулю.

Обчислимо добуток $h(t) = \sum_{i=0}^{\infty} c_i t^i = f(t)g(t)$:

$c_0 = u_0;$

$c_1 = u_1 - a_1 u_0;$

$c_2 = u_2 - a_1 u_1 - a_2 u_0;$

....

$c_r = u_r - a_1 u_{r-1} - \ldots - a_r u_0 = 0;$

....

$c_n = u_n - a_1 u_{n-1} - \ldots - a_r u_{n-r} = 0.$

Отже, степеневий ряд $h(t)$ є також багаточленом, а породна функція $f(t)$ є результатом ділення багаточленів: $f(t) = h(t)/g(t)$.

Існує техніка розкладання таких виразів «на найпростіший дріб», за допомогою якої можна обчислити коефіцієнти дробу $\frac{h(t)}{g(t)}$ у остаточному вигляді як функції від номера коефіцієнта. Це і буде повним формальним розв'язком поставленої задачі.

Приклад. Нехай $u_0 = u_1 = 1$ і $u_n = u_{n-1} + u_{n-2}$, $n \geq 2$. Така послідовність називається *числами Фібоначчі* і виглядає так: 1,1,2,3,5,8,13,21... . Знайдемо n-е число Фібоначчі як функцію від n. Маємо: $g(t) = 1 - a_1 t - a_2 t^2$ і, продовжуючи збереження позначень, $f(t) = u_0 + u_1 t^2 + \ldots + u_n t^n + \ldots$. Звідси $f(t)g(t)$ має вигляд: $f(t)g(t) = u_0 - (u_1 - u_0)t = 1$. Отже, $f(t) = \frac{1}{1 - t - t^2}$.

Відзначимо, що: $1 - t - t^2 = -(t^2 + t - 1) = -(t - t_1)(t - t_2)$,

де $t_1 = -\frac{\sqrt{5}+1}{2}, t_2 = \frac{\sqrt{5}-1}{2}$.

Отже, $f(t) = -\frac{1}{t^2 + t - 1} = \frac{1}{t_2 - t_1}\left(\frac{1}{t - t_1} - \frac{1}{t - t_2}\right) = \frac{1}{\sqrt{5}}\left(\frac{1}{t_2}\frac{1}{1 - \frac{t}{t_2}} - \frac{1}{t_1}\frac{1}{1 - \frac{t}{t_1}}\right)$,

а відповідно до правила ділення формальних степеневих рядів:

$$j(t)=\frac{1}{\sqrt{5}}\left(\frac{1}{t_2}\sum_{k=0}^{\infty}\left(\frac{t}{t_2}\right)^k-\frac{1}{t_1}\sum_{k=0}^{\infty}\left(\frac{t}{t_1}\right)^k\right)=\frac{1}{\sqrt{5}}\left(\sum_{k=0}^{\infty}\left(\frac{1}{t_2^{k+1}}-\frac{1}{t_1^{k+1}}\right)t^k\right)=$$

$$=\frac{1}{\sqrt{5}}\sum_{k=0}^{\infty}\frac{t_1^{k+1}-t_2^{k+1}}{(t_1t_2)^{k+1}}t^k=\frac{1}{\sqrt{5}}\sum_{k=0}^{\infty}(-1)^{k+1}\left(t_1^{k+1}-t_2^{k+1}\right)t^k.$$

Звідси випливає, що $u_k=\frac{(-1)^{k+1}}{\sqrt{5}}\left(t_1^{k+1}-t_2^{k+1}\right),\quad k=0,1,2,\ldots$.

5 КОНТРОЛЬНІ ЗАПИТАННЯ І ЗАВДАННЯ

5.1 Елементи загальної алгебри

5.1.1 Чи є алгебраїчними такі операції :

a) ділення на множині раціональних чисел,
b) ділення на множині цілих чисел,
c) віднімання на множині цілих чисел,
d) віднімання на множині натуральних чисел,
e) обчислення кореня на множині дійсних чисел,
f) обчислення кореня на множині дійсних чисел більших за одиницю,
g) обчислення кореня на множині дійсних чисел більших за два?

5.1.2 Для яких p $<\{1, 2,\ldots, p\};\ (\cdot \bmod (p+1))>$ є алгебраїчною системою.

5.1.3 Довести, що наведені алгебраїчні системи ізоморфні:

a) $<\mathbf{N}; \leq>$ і $<\mathbf{N} \setminus \{1\}; \leq>$.
b) $<\{0, 2,\ldots, 2\cdot n, \ldots\}; +; \leq>$ і $<\{0, 3,\ldots, 3\cdot n, \ldots\}; +; \leq>$.

5.1.4 Довести, що наведені алгебраїчні системи не є ізоморфними:

a) $<\mathbf{N}; +>$ і $<\mathbf{Z}; +>$;
b) $<\mathbf{N}; +; \leq>$ і $<\mathbf{N}; -; \leq>$;
c) $<\mathbf{R}; +>$ і $<\mathbf{Q}; +>$;
d) $<\mathbf{Z}; (+ \bmod 3)>$ і $<\mathbf{Z}; (\cdot \bmod 3)>$;
e) $<\{0, 2, \ldots, 2\cdot n, \ldots\}; \cdot; \leq>$ і $<\{0, 3, \ldots, 3\cdot n, \ldots\}; \cdot; \leq>$.

5.1.5 Чи є система A підсистемою алгебраїчної системи B:

a) $A = B = <\mathbf{N}; +>$;
b) $A = <\mathbf{N}; +>$, $B = <\mathbf{Z}; +>$;
c) $A = <\mathbf{N}\setminus\{1\}; +>$, $B = <\mathbf{N}; +>$;

d) $A = \langle \mathbf{N}; \cdot \rangle$, $B = \langle \mathbf{N}; + \rangle$;
e) $A = \langle \mathbf{N}; + \rangle$, $B = \langle \mathbf{N} \setminus \{1\}; + \rangle$;
f) $A = \langle \{1, 2, ..., n\}; ; <, \leq \rangle$, $B = \langle \{1, 2, ..., n\}; ; <, \geq \rangle$?

5.1.6 Для яких p алгебраїчна система $\langle \{0, 1, ..., p\text{-}1\}; (+ \bmod p); \leq \rangle$ є ізоморфною алгебраїчним системам:

a) $\langle \{1, 2, ..., p\}; (\cdot \bmod (p+1)); \leq \rangle$,
b) $\langle \{1, 2, ..., p\}; ((\cdot \bmod p)+1); \leq \rangle$.

5.1.7 Перевірити, ізоморфні чи ні алгебраїчні системи:

a) $\langle \mathbf{Z}; + \rangle$ і $\langle \mathbf{Z}; - \rangle$;
b) $\langle \mathbf{Z}; +; \leq \rangle$ і $\langle \mathbf{Z}; +; \geq \rangle$;
c) $\langle \mathbf{Z}; -; \leq \rangle$ і $\langle \mathbf{Z}; -; \geq \rangle$;
d) $\langle \mathbf{Z}; \cdot; \leq \rangle$ і $\langle \mathbf{Z}; \cdot; \geq \rangle$.

5.1.8 Чи існує в наведених нижче прикладах гомоморфізм алгебраїчної системи A в алгебраїчній системі B :

b) $A = \langle \mathbf{N}; + \rangle$, $B = \langle \mathbf{N}; \cdot \rangle$;
c) $A = \langle \mathbf{N}; + \rangle^2$ (множина пар натуральних чисел з покоординатним додаванням), $B = \langle \mathbf{N}; \cdot \rangle$;
d) $A = \langle \mathbf{N}; \cdot \rangle$, $B = \langle \mathbf{N}; + \rangle$;
e) $A = \langle \mathbf{N}; + \rangle$, $B = \langle \mathbf{N} \setminus \{1\}; \cdot \rangle$;
f) $A = \langle \{1, 2, ..., n\}; ; <, \leq \rangle$, $B = \langle \{1, 2, ..., n\}; ; <, \geq \rangle$;
g) $A = \langle \{1, 2, ..., n\}; ; <, \leq \rangle$, $B = \langle \{1, 2, ..., n\}; ; \leq, \geq \rangle$?

5.1.9 Знайти кількість ізоморфізмів алгебраїчної системи

$\langle \mathbf{R}; +; \leq \rangle$ на $\langle \mathbf{R}_+ \setminus \{0\}; \cdot ; \leq \rangle$.

5.1.10 Знайти кількість гомоморфізмів :

a) алгебри $\langle \mathbf{Z}; + \rangle$ у себе;
b) моделі $\langle \{1, 2, ..., n\}; \leq \rangle$ у $\langle \{1, 2, ..., n\}; \geq \rangle$.

5.1.11 Знайти кількість автоморфізмів моделі :

a) $\langle \{1,2,3,4,5,6\}; r \rangle$, де r – відношення взаємної непростоти;
b) $\langle \{1, 2, ..., n\}; \neq \rangle$.

5.1.12 Знайти кількість автоморфізмів алгебр:

a) $\langle \mathbf{N}; + \rangle$;
b) $\langle \mathbf{N}; \cdot \rangle$;
c) $\langle \mathbf{N}; + \rangle^2$;
d) $\langle \{0, 1, ..., p\text{-}1\}; (+ \bmod p) \rangle$, де p - простої число;
e) $\langle \{0, 1, ..., 5\}; (+ \bmod 6) \rangle$.

5.1.13 Знайти граф з мінімальним числом вершин, не менше двох, що має тільки тривіальний автоморфізм.

5.1.14 Знайти простий граф з мінімальним числом вершин не менше двох, що має тільки тривіальний автоморфізм.

5.1.15 Чи будуть гратками :

a) $< \{1, ..., n\}^n$; min, max $>$;
b) $<$ **N**; НЗД, НЗК $>$;
c) $< \{0, 1\}^n$; min, max $>$;
d) $<$ **Z**; min, max $>$?

5.1.18 Чи будуть булевими гратками :

e) $< \{0, 1, ..., n\text{-}1\}$; (+ mod n), ($\cdot$ mod n) $>$;
f) $< \{0, 1, ..., n\text{-}1\}$; max, min $>$;
g) $< \{0, 1, i, i+1\}; +, \cdot >$?

5.1.17 Знайти замикання множин на булевих гратках

$< \{(x_1, ..., x_n) | x_i \in \{0, 1\} \}$; max, min $>$:

a) $\{(x_1, ..., x_n) | S_i x_i = 1\}$;
b) $\{(x_1, ..., x_n) | S_i x_i = 2\}$;
c) $\{(x_1, ..., x_n) | S_i x_i$ і $n\text{-}1\}$;
d) $\{(x_1, ..., x_n) | x_1 = 0\}$;
e) $\{(0, ..., 0), (1, ..., 1), (0, 1, 0, 1, ..., 0, 1) \}$;
f) $\{ (0, 1, 0, 1, ..., 0, 1), (1, 1, 0, 1, ..., 0, 1) \}$.

5.1.18 Чи будуть булевими алгебрами :

a) $< \{0, 1, ..., n\text{-}1\}$; (+ mod n), ($\cdot$ mod n), $n\text{-}x$, 0, $n\text{-}1$ $>$;
b) $< \{0, 1, ..., n\text{-}1\}$; max, min, $n\text{-}x$, 0, $n\text{-}1$ $>$;
c) $< \{0, 1, i, i+1\}; +, \cdot, i+1\text{-}x, 0, i+1 >$?

5.1.19 У булевій алгебрі з носієм {(0,0,0,0), (0,0,0,1),..., (1,1,1,1)} і операціями, що визначені як + – як sup, $\cdot$ – як inf, $\neg x$ – (1,1,1,1) - x, 0 – (0,0,0,0), 1 – (1,1,1,1) вказати:

a) всі атоми;
b) усі суми пар атомів;
c) усі суми трійок атомів;
d) всі елементи, що мають вигляд: $a \cdot \neg b + b \cdot \neg a$, де a і b - атоми;
e) всі елементи, що мають вигляд: $(a + b) \cdot (\neg a + \neg b$, де a і b є елементами булевої алгебри.

5.2 Булеві функції

5.2.1 Вкажіть фіктивні змінні функції $f(x, y, z)$, що задана векторами:

a) f = (1,1,1,1,0,0,0,0);
b) f = (0,0,1,1,0,0,1,1);
c) f = (0,0,1,1,1,1,0,0).

5.2.2 Знайдіть кількість функцій n змінних, що приймають значення 1 рівно на одному наборі.

5.2.3 Знайдіть кількість функцій n змінних, що приймають на протилежних наборах однакові значення.

5.2.4 Побудуйте таблицю функції f:

a) $f(x, y) = (x \supset y)\ \&\ (y \supset x)$;
b) $f(x, y, z) = (x\ \&\ y) \oplus (x\ \&\ z) \oplus (y\ \&\ z)$;
c) $f(x, y, z) = (x\ \&\ y) \vee (x\ \&\ z) \vee (y\ \&\ z)$;
d) $f(x, y, z) = (x\ \&\ \bar{y}) \vee (x\ \&\ \bar{z}) \vee (y\ \&\ \bar{z})$;
e) $f(x, y) = x \mid (x \mid y)$;
f) $f(x, y) = (x \mid y) \mid (x \mid y)$.

5.2.5 За функціями f і g побудувати векторне задання функції h:

a) $f = (0,0,1,0)$, $g = (0,1,0,0)$, $h(x, y, z) = f(g(x, y), z)$;
b) $f = (1,1,1,0)$, $g = (1,0,1,1)$, $h(x, y, z) = f(x, y)\ \&\ g(z, y)$;
c) $f = (1,1,1,0)$, $g = (1,0,0,0)$, $h(x, y, z) = f(g(x, y), g(y, z))$.

5.2.6 Перевірте, чи справедливі такі співвідношення:

a) $x \vee (y \equiv z) = (x \vee y) \equiv (x \vee z)$;
b) $x \rightarrow (y \equiv z) = (x \rightarrow y) \equiv (x \rightarrow z)$;
c) $x\ \&\ (y \equiv z) = (x\ \&\ y) \equiv (x\ \&\ z)$;
d) $x \rightarrow (y \vee z) = (x \rightarrow y) \vee (x \rightarrow z)$;
e) $x \rightarrow (y\ \&\ z) = (x \rightarrow y)\ \&\ (x \rightarrow z)$;
f) $x \oplus (y\ \&\ z) = (x \oplus y)\ \&\ (x \oplus z)$;
g) $x \rightarrow (y \rightarrow z) = (x \rightarrow y) \rightarrow (x \rightarrow z)$.

5.2.7 Знайдіть функцію, двоїсту до функції $(x\ \&\ (y \vee z)) \oplus x$.

5.2.8 Доведіть теорему про розкладання в кон'юнкцію використовуючи теорему про розкладання в диз'юнкцію і принцип двоїстості.

5.2.9 Найдіть ДДНФ таких функцій:

a) $f_1(x, y, z) = (1,1,0,1,0,0,0,0)$,
b) $f_2(x, y, z) = x\ \&\ y\ \&\ z$,
c) $f_3(x, y, z) = x \oplus y \oplus z$,
d) $f_4(x, y, z) = \overline{((x\ \&\ y) \supset z)}$.

5.2.10 Знайдіть ДКНФ таких функцій:

a) $f_1(x, y, z) = (1,1,0,1,0,0,0,0)$,
b) $f_2(x, y, z) = x \vee y \vee z$,
c) $f_3(x, y, z) = x \oplus y \oplus z$.

5.2.11 Знайдіть кількість диз'юнктивних членів у досконалих диз'юнктивних нормальних формах наступних функцій:

a) $f_1(x_1,\ldots,x_n) = x_1 \oplus \ldots \oplus x_n$,

b) $f_2(x_1,\ldots,x_n) = x_1 \vee \ldots \vee x_n$,

c) $f_3(x_1,\ldots,x_n) = (x_1 \vee \ldots \vee x_n) \,\&\, (\bar{x}_1 \vee \ldots \vee \bar{y}_n)$.

5.3 Графи

5.3.1 Теоретико-множинне введення

5.3.1.1 Доведіть рівність: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, де A, B, C - деякі множини.

5.3.1.2 Доведіть рівність: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, де A, B, C - деякі множини.

5.3.1.3 Чи вірна рівність: $A \setminus (B \cup C) = (A \setminus B) \cap (A \setminus C)$, де A, B, C - деякі множини?

5.3.1.4 Чи вірна рівність: $A \setminus (B \cup C) = (A \setminus B) \cup (A \setminus C)$, де A, B, C - деякі множини?

5.3.1.5 Що є доповненням до множини парних чисел у множині натуральних чисел ?

5.3.1.6 Що є доповненням до множини $\{1,3,5\}$ у множині $\{1,2,3,4,5,6\}$?

5.3.1.7 Що є доповненням до множини $\{1,3,5\}$ у множині $\{1,3,5\}$?

5.3.1.8 Дані дві множини $A = \{1,2,3,4\}$, $B = \{2,3,6,8,9\}$; запишіть $A \times B$.

5.3.1.9 Дані дві множини $A = \{1,2,3,4\}$, $B = \{2,3,6,8,9\}$; запишіть $B \times A$.

5.3.1.10 Дана множина $A = \{3,2,1,0,7,6\}$. Запишіть $A \times A$.

5.3.1.11 Дана множина $A = \{2,5,3,4,8,1\}$. Запишіть її діагональ.

5.3.1.12 Наведіть приклад двох різних рефлексивних відношень на множині $A = \{2,5,3,4,8,1\}$.

5.3.1.13 Наведіть приклад чотирьох різних рефлексивних відношень на множині $A = \{2,5,3,4,8,1\}$.

5.3.1.14 Наведіть приклад трьох різних відношень на множині $A = \{2,5,3,4,8,1\}$, що не є рефлексивними.

5.3.1.15 Наведіть приклад двох різних симетричних відношень на множині $\{1,2,4,6,7,0,10\}$.

5.3.1.16 Наведіть приклад двох різних відношень на множині $\{1,2,4,6,7,0,10\}$, що не є симетричними.

5.3.1.17 Наведіть приклад двох різних антисиметричних відношень на множині $\{1,2,4,6,7,0,10\}$.

5.3.1.18 Наведіть приклад двох різних відношень на множині $\{1,2,4,6,7,0,10\}$, що не є антисиметричними.

5.3.1.19 Наведіть приклад двох різних транзитивних відношень на множині $\{1,2,4,6,7,0,10\}$.

5.3.1.20 Наведіть приклад двох різних відношень на множині

$\{1.2.4.6.7.0.10\}$, що не є транзитивними.

5.3.1.21 Нехай R - деяке відношення на множини M: зіставимо відношенню R таку таблицю

рефлексивность	*симетричність*	*транзитивність*	*антисиметричність*
*	*	*	*

Замість символу "*" поставимо "+" якщо властивість, зазначена над "*", є у відношенні R, і поставимо "-" якщо властивість, зазначена над "*", у відношенні R відсутня. Легко зрозуміти, що усього існує 16 варіантів заповнення цієї таблиці:

- | - | - | -	- | - | - | +	- | + | + | -	+ | + | - | +
+ | - | - | -	+ | + | - | -	- | + | - | +	+ | - | + | +
- | + | - | -	+ | - | + | -	- | - | + | +	- | + | + | +
- | - | + | -	+ | - | - | +	+ | + | + | -	+ | + | + | +

Виберіть за власним бажанням множину M і побудуйте два приклади відношення R для кожної таблиці в цій множини M, причому множина M та сама у всіх шістнадцятьох прикладах.

5.3.1.22 Наведіть приклад еквівалентності і приклад відношення, що не є еквівалентністю.

5.3.1.23 Наведіть приклад часткового порядку і приклад відношення, що не є частковим порядком.

5.3.1.24 Як охарактеризувати відношення, що є еквівалентністю і частковим порядком одночасно?

5.3.1.25 Наведіть приклад множини і двох різних еквівалентностей на цій множині.

5.3.1.26 Наведіть приклад множини і двох різних часткових порядків на цій множині.

5.3.2. Визначення графа і його основних характеристик

5.3.2.1 Є множина $A = \{1.2.3.4.5.6\}$. Записати множину $V(A)$.

5.3.2.2 Скільки вершин і ребер у графа

$$\Gamma = [\ A = \{1.2.3.4.5.6\},\quad B = \{(1.2).(1.3).(1.4).(3.2).(3.5).(3.6).(4.6)\}\ ]$$

і як він виглядає графічно?

5.3.2.3 З умови 5.3.2.2 навести суміжні і несуміжні вершини.

5.3.2.4 З умови 5.3.2.2 навести суміжні і несуміжні ребра.

5.3.2.5 З умови 5.3.2.2 навести інцидентні і неінцидентні об'єкти.

5.3.2.6 Чому дорівнює локальний степінь графа 5.3.2.2 вершині «1»?

5.3.2.7 Чи є серед вершин графа 5.3.2.2 ізольовані?

5.3.2.8 Чому дорівнює степінь вершини «6» графа з задачі 2.2?

5.3.2.9 Наведіть приклад графа, серед локальних степенів якого будуть наперед задані числа.

5.3.2.10 Скільки ребер у повному графі на семи і восьми вершинах?

5.3.2.11 Скільки вершин непарного степеня у графі 5.3.2.2? Скільки там вершин парного степеня? Навести приклад графа, у якому число парних локальних степенів дорівнює п'яти. Скільки у Вашому прикладі непарних локальних степенів?

5.3.2.12 Маємо два графи: $\Gamma_1 = [A_1, B_1]$, $\Gamma_2 = [A_2, B_2]$, де $A_1 = \{1,2,3,4,5\}$, $A_2 = = \{1,3,5\}$, $B_1 = \{(1,2),(1,3),(1,5),(2,3),(2,5),(3,4),(3,5)\}$, $B_2 = \{(2,5),(2,3),(3,5),(1,5)\}$. Чи є один з них підграфом іншого?

5.3.2.13 Наведіть приклад двох графів, жоден з який не є підграфом іншого.

5.3.2.14 У задачах 5.3.2.2 і 5.3.2.12 наведено конкретні графи. Записати для кожного з них матрицю суміжностей і матрицю інциденцій.

5.3.2.15 Дані такі матриці:

0	0	1	0	1	0	1
0	0	0	1	0	1	1
1	0	0	1	1	0	0
0	1	1	0	0	1	1
1	0	1	0	0	0	1
0	1	0	1	0	0	1
1	1	0	1	1	1	0

0	1	0	1	0	1	0	1
1	0	1	0	1	0	0	1
1	0	0	1	1	1	0	0
0	0	1	0	1	0	1	0
0	0	0	0	0	1	0	0
1	0	0	0	1	0	0	1
1	1	1	1	0	0	0	1
1	1	1	1	0	0	0	0

1	0	0	1	0	1	0
1	1	1	0	1	0	1
0	0	0	0	0	0	0
1	0	1	0	1	0	0

1	1	1	1	0	0
1	0	0	0	0	0
0	1	0	0	1	1
0	0	1	0	1	0
0	0	0	1	0	1

1	1	1	1	1	0	0
0	0	0	1	0	1	0
0	1	0	0	1	1	1
0	0	1	0	0	0	1
1	0	0	0	0	0	0

1	1	1	1	1	1
1	0	1	0	1	0
0	1	0	0	0	1
0	0	0	1	0	0
0	0	0	0	0	0

Встановити, які з них є матрицями суміжностей, які - матрицями інциденцій і які не є ні тими, ні іншими.

5.3.2.16 Встановити, що якщо при ізоморфізмі f двох графів вершина a переходить у вершину $f(a)$, то степені обох вершин рівні.

5.3.2.17 Навести приклад графів ізоморфних і графів неізоморфних.

5.3.2.18 Як організувати перевірку двох графів на ізоморфізм?

5.3.3 Зв'язність та інші властивості графів

5.3.3.1 Маємо граф з множиною вершин $\{1,2,3,4,5,6,7\}$. Нижче у таблиці для кожної вершини перераховані суміжні з нею вершини. Навести приклад:

а) шляху у цьому графі, що не є ланцюгом;
б) ланцюга у цьому графі, що не є простим;
в) замкненого шляху, що не є циклом;
г) циклу, що не є простим;
д) простого циклу.

Вершина	Суміжні з нею
1	4,5,7
2	3,6
3	2,4,5,7
4	1,3
5	1,3,6
6	5,2,7
7	1,3,6

5.3.3.2 Навести приклад графа, у якому буде пара незв'язних вершин і пара зв'язних вершин.

5.3.3.3 Навести приклад графа з трьома зв'язними компонентами, приклад зв'язного графа і приклад графа, у якому видалення однієї вершини збільшує число зв'язних компонентів.

5.3.3.4 Навести приклад графа, у якому видалення деяких двох ребер збільшує число зв'язних компонентів, однак видалення будь-якого одного ребра число зв'язних компонентів не змінює.

5.3.3.5 Навести приклад графа, що є деревом, і графа, що ним не є.

5.3.3.6 Навести приклад двох дерев з однаковими кількостями вершин, які були б не ізоморфними графами.

5.3.3.7 Маємо зв'язний граф. Чи можна шляхом видалення з нього деяких ребер одержати дерево?

5.3.3.8 Вершину степеня один прийнято називати висячою. Чи обов'язково у кожному дереві є висяча вершина?

5.3.3.9 У дереві 12 вершин. Скільки в ньому ребер?

5.3.3.10 Маємо граф, що є простим ланцюгом. Чи є він деревом?

5.3.3.11 Маємо дерево. Деяке його ребро піддане операції розбиття. Чи може отриманий граф не бути деревом?

5.3.3.12 Перевірити чи буде будь-яка пара зв'язних вершин з'єднана простим ланцюгом.

5.3.3.13 Перевірити чи будуть у дереві будь-які дві вершини з'єднані єдиним простим ланцюгом.

5.3.3.14 Задано повний граф на 15 вершинах. У результаті послідовного видалення декількох ребер від нього залишилося дерево. Скільки ребер видалили?

5.3.3.15 У дереві видалили вершину, степінь якої був більшим одиниці. Чи буде зв'язним граф, що залишився в результаті?

5.3.4 Ейлерові і Гамільтонові графи

5.3.4.1 Навести приклад двох неізоморфних ейлерових графів.

5.3.4.2 Навести приклад неізоморфних графів, що не є ейлеровими.

5.3.4.3 Маємо повний граф на n вершинах. Чи є він ейлеровим?

5.3.4.4 Є повний граф на 16 вершинах. Яке мінімальне число простих ланцюгів у графі, що у сукупності містять усі його ребра і вершини?

5.3.4.5 Чи існує ейлерів граф, що має висячу вершину?

5.3.4.6 Навести неейлеровий граф, усі локальні степені якого парні.

5.3.4.7 Навести приклад зв'язного графа, що не є ейлеровим.

5.3.4.8 Є ейлерів граф. Як реально знайти ейлерів цикл?

5.3.4.9 Навести приклади гамільтонового і негамильтонового графа.

5.3.4.10 Навести приклади графа Дірака і графа, що ним не є.

5.3.4.11 Навести приклад графа Оре і приклад графа, що ним не є.

5.3.4.12 Побудувати функцію Поша для таких графів:

а) повний граф на семи вершинах;

б) простий ланцюг на шести вершинах;

в) простий цикл на восьми вершинах;

г) цикл з єдиним самоперетинанням.

5.3.4.13 Навести приклад графа Поша і приклад графа, що ним не є.

5.3.4.14 Навести гамільтоновий граф, що не є графом Дірака.

5.3.4.15 Навести гамільтоновий граф, що не є графом Оре.

5.3.4.16 Навести гамільтоновий граф, що не є графом Поша.

5.3.4.17 Навести приклад графа Оре, що не є графом Дірака.

5.3.4.18 Навести приклад графа Поша, що не є графом Дірака.

5.3.4.19 Чи є граф Дірака графом Оре?

5.3.4.20 Чи є граф Дірака графом Поша?

5.3.5 Теорема Менгера

5.3.5.1 Навести приклад (u,v)-відокремлювальної множини вершин.

5.3.5.2 Навести приклад (u,v)-відокремлювальної множини ребер.

5.3.5.3 Навести приклад вершинно-неперетинанних (u,v)-ланцюгів.

5.3.5.4 Навести приклад реберно-неперетинанних (u,v)-ланцюгів.

5.3.5.5 Навести приклад чотирьох неперетинанних (u,v)-ланцюгів.

5.3.5.6 Навести приклад двох (u,v)-відокремлювальних множин, одна з яких має мінімально можливе число вершин, а інша складається з більшого числа вершин, але ні однієї вершини з нього видалити не можна, залишаючи його (u,v)-відокремлювальним.

5.3.5.7 Навести приклад двох множин вершино-неперетинанних (u,v)-ланцюгів, в одному з яких максимально можливе число ланцюгів, а в іншому - число ланцюгів менше, але до нього не можна додати жодного

(u, v)-ланцюга так, щоб знову вийшла множина вершино-неперетинанних (u, v)-ланцюгів.

5.3.5.8 Навести приклад двох (u, v)-відокремлювальних множин ребер, одна з яких має мінімально можливе число елементів, а в іншій - число елементів більше, але якщо з нього хоч одне ребро забрати, то отримана множина уже не буде (u, v)-відокремлювальною.

5.3.5.9 Навести приклад системи множин і системи різних представників для неї.

3.5.10 Навести приклад системи множин, у якої немає системи різних представників.

5.3.5.11 У бінарній матриці навести приклад системи незалежних клітинок.

5.3.5.12 У бінарній матриці навести приклад системи незалежних клітин, що містить максимальну кількість клітинок.

5.3.5.13 Чи можна навести приклад незалежних клітинок у бінарній матриці такий, що до сукупності цих клітинок не можна додати ні однієї клітинки так, щоб знову вийшов набір незалежних клітинок, однак разом з тим, кількість клітинок у цьому наборі не максимальна?

5.3.5.14 У бінарній матриці навести приклад системи покривних ліній.

5.3.5.15 У бінарній матриці навести приклад системи покривних ліній, що містить мінімальне число ліній.

5.3.5.16 У бінарній матриці навести приклад двох систем покривних ліній: в одній системі число ліній повинно бути мінімальним, а в іншій системі число ліній повинно бути хоча б на одну більше, однак, якщо з останньої системи будь-яку лінію забрати, то ті, що залишилися не будуть складати систему покривних ліній.

5.3.6 Паросполучення і дводольні графи

5.3.6.1 Навести приклади дводольного і не дводольного графів.

5.3.6.2 Навести приклад дводольного графа, записати його матрицю і матрицю суміжностей.

5.3.6.3 Навести приклад паросполучення і приклад набору ребер, що не є паросполученням.

5.3.6.4 Навести приклад максимального паросполучення .

5.3.6.5 Навести приклад найбільшого паросполучення .

5.3.6.6 Навести приклад максимального паросполучення, що не є найбільшим.

5.3.6.7 Навести приклад двох різних максимальних паросполучень в одному графі.

5.3.6.8 Навести приклад двох різних найбільших паросполучень в одному графі.

5.3.6.9 Чи є дводольним графом дерево?

5.3.6.10 Чи є дводольним графом повний граф?

5.3.6.11 Чи є дводольним графом простий ланцюг?

5.3.6.12 Навести приклад трьох різних максимальних паросполучень в одному дводольному графі.

5.3.6.13 Нехай у графі зафіксовано паросполучення і простий ланцюг з такими властивостями: - його перше і останнє ребра паросполученню не належать; - з будь-яких двох послідовних ребер одне належить паросполученню, а друге - ні. Чи буде у цій ситуації дане паросполучення найбільшим?

5.3.6.14 Для наведених нижче матриць дводольних графів знайти найбільше паросполучення.

1	1	1	1	1	1
1	0	1	1	1	1
1	0	1	0	1	1
0	0	1	0	1	1
1	0	1	0	1	1
1	0	1	0	1	1

1	1	0	1	1	0
1	1	1	0	0	1
0	0	1	1	1	1
1	1	1	0	0	0
1	1	1	0	0	0
1	1	0	0	0	0

0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	0	0	0	0
1	1	1	1	0	0	1	0
1	1	1	1	0	1	0	1
1	1	1	1	0	1	0	0
1	1	1	1	0	1	0	0

1	1	0	0	0
0	1	1	0	0
0	0	1	1	0
1	0	0	0	1
0	0	0	1	0

0	1	0	0	1
0	1	1	0	0
0	0	1	1	0
1	0	0	0	1
0	0	0	1	0

1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	0

5.3.6.15 Дані матриці ефективностей n робітників на n місцях. Знайти розв'язання задачі про призначення на вузькі місця.

2	4	5	2	3	4
3	4	6	3	4	5
6	5	4	3	6	5
4	6	7	6	5	4
3	4	5	6	5	4
2	5	6	7	9	8
5	6	7	8	7	6

3	4	4	5	4	2
5	4	3	4	6	7
6	7	6	7	6	1
5	4	3	4	5	6
8	7	6	5	4	6
5	6	7	8	4	3

8	7	6	5	4	5
6	7	8	7	6	5
4	5	6	7	8	7
6	5	4	5	6	7
5	7	8	9	8	7
6	6	7	7	9	5

5.3.7 Частково-упорядковані множини і мінімальні розбивки

5.3.7.1 Навести приклад різних часткових порядків на одній множині.

5.3.7.2 Навести приклад трьох різних ланцюгів у частково-упорядкованій множині.

5.3.7.3 Навести приклади мінімальної і немінімальної ланцюгової розбивки.

5.3.7.4 Навести приклад двох різних антиланцюгів у частково-упорядкованій множині.

5.3.7.5 Навести приклад частково-упорядкованої множини і побудувати відповідний їй дводольний граф.

5.3.7.6 Знайти мінімальну ланцюгову розбивку таких частково упорядкованих множин:

а) $A = \{1,2,3,4,5,6,7,8\}$, $R = \{(1,2),(2,3),(1,3),(4,5),(5,6),(4,6)\} \cup \Delta(A)$;

б) $A = \{1,2,3,4,5\}$, $R = \{(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(4,5)\} \cup \Delta(a)$;

в) $A = \{1,2,3,4,5\}$, $R = \{(1,3),(1,5),(2,4),(3,5),(4,2),(5,2),(5,4)\} \cup \Delta(A)$.

г) $A=\{1,2,3,4,5,6\}$, $\{(2,3),(2,4),(2,5),(2,6),(3,5),(3,6),(4,6),(5,6)\} \cup \Delta(A)$.

5.3.8 Планарні і плоскі графи

5.3.8.1 Навести приклад графа планарного і графа непланарного.

5.3.8.2 Навести приклад графа планарного, але не плоского.

5.3.8.3 Навести приклад графа плоского і графа неплоского.

5.3.8.4 Навести приклад плоского графа і перелічити всі його грані.

5.3.8.5 Навести приклад плоского графа, в якого рівно п'ять граней.

5.3.8.6 Навести приклад плоского графа, всі грані якого трикутні.

5.3.8.7 Навести приклад плоского графа, всі грані якого чотирикутні.

5.3.8.8 Чи є планарним повний граф?

5.3.8.9 За яких умов буде планарним дводольний граф?

5.3.8.10 Чи буде планарним дерево?

5.3.8.11 Чи може бути планарним незв'язний граф?

5.3.8.12 Маємо географічну карту, де різні країни розфарбовані різними кольорами. Побудовано граф, кожна вершина якого відповідає країні на карті, а ребрами з'єднані ті і тільки ті його вершини, яким на карті відповідають країни, що граничать між собою. Чи буде цей граф плоским?

5.3.9 Розфарбування графів

5.3.9.1 Навести приклад розфарбування графа.

5.3.9.2 Навести приклад графа з хроматичним числом один.

5.3.9.3 Навести приклад графа з хроматичним числом п'ять.

5.3.9.4 Для кожного наперед заданого натурального числа n навести приклад графа з хроматичним числом n.

5.3.9.5 Чому дорівнює хроматичне число дерева?

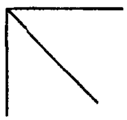
5.3.9.6 Нехай Г - деякий граф, $\chi(\Gamma)$ - його хроматичне число, а $\Delta(\Gamma)$ - його максимальний локальний степінь. Наскільки великим і наскільки малим може бути число $\Delta(\Gamma)+1-\chi(\Gamma)$?

5.3.9.7 Чому дорівнює старший коефіцієнт хроматичного багаточлена графа?

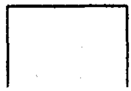
5.3.9.8 Чому дорівнює вільний член хроматичного багаточлена ?

5.3.9.9 Для наведених нижче графів знайти їх хроматичні багаточлени.

а)

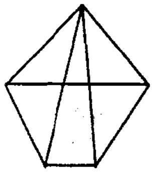


б)

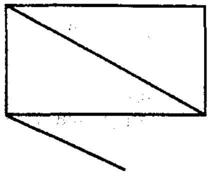


в)

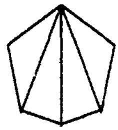
г)



д)



е)



5.3.10 Зважені графи

5.3.10.1 У наведених зважених графах вказати цикли і їх ваги.

а)

0	2	1	4	5	2
2	0	3	2	1	4
1	3	0	5	6	4
4	2	5	0	2	1
5	1	6	2	0	6
2	4	4	1	6	0

б)

0	0	1	0	2	3
0	0	0	4	3	0
1	0	0	2	0	8
0	4	2	0	5	7
2	3	0	5	0	0
3	0	8	7	0	0

в)

0	2	4	3	0	3	0
2	0	6	0	9	5	0
4	6	0	4	5	0	4
3	0	4	0	3	4	5
0	9	5	3	0	5	2
3	5	0	4	5	0	6
0	0	4	5	2	6	0

5.3.10.2 Знайти найкоротші маршрути з першої вершини в усі інші вершини зваженого графа:

а)

0	5	8	4	0	8	6	5	3	7
5	0	5	2	8	6	1	0	6	3
8	5	0	4	5	6	8	2	1	3
4	2	4	0	0	4	7	0	4	3
0	8	5	0	0	2	1	5	6	3
8	6	6	4	2	0	3	5	0	1
6	1	8	7	1	3	0	7	6	9
5	0	2	0	5	5	7	0	4	0
3	6	1	4	6	0	6	4	0	6
7	3	3	3	3	1	9	0	6	0

б)

0	6	5	9	0	4	2	0	5	0
6	0	0	5	8	6	9	4	2	6
5	0	0	4	6	9	2	1	6	4
9	5	4	0	5	5	4	3	3	1
0	8	6	5	0	4	3	2	4	1
4	6	9	5	4	0	2	6	3	1
2	9	2	4	3	2	0	4	6	4
0	4	1	3	2	6	4	0	9	9
5	2	6	3	4	3	6	9	0	6
0	6	4	1	1	1	4	9	6	0

в)

0	2	0	7	8	0	5	9
2	0	5	6	8	3	1	4
0	5	0	3	1	4	2	7
7	6	3	0	1	9	4	5
8	8	1	1	0	9	0	7
0	3	4	9	9	0	9	8
5	1	2	4	0	9	0	5
9	4	7	5	7	8	5	0

г)

0	1	8	9	8	2	0	5
1	0	1	9	4	5	0	8
8	1	0	1	9	0	7	0
9	9	1	0	1	1	0	5
8	4	9	1	0	2	3	8
2	5	0	1	2	0	4	1
0	0	7	0	3	4	0	5
5	8	0	5	8	1	5	0

д)

0	6	5	7	0	5	0	6
6	0	9	8	3	7	8	0
5	9	0	4	6	7	1	8
7	8	4	0	2	7	4	3
0	3	6	2	0	4	3	2
5	7	7	7	4	0	9	0
0	8	1	4	3	9	0	5
6	0	8	3	2	0	5	0

5.3.10.3 Навести приклад дерева у кожному з графів 5.3.10.2.

5.3.10.4 Знайти найлегше дерево за алгоритмом Краскала до кожного з графів 5.3.10.2.

5.3.10.5 Маємо зв'язний дводольний граф, одна з часток якого складається з 10 вершин, а інша - 12 вершин. Скільки вершин і ребер у дереві?

5.3.10.6 Навести приклад зваженого графа, у якому є два найлегших дерева для загального числа дерев не менше трьох.

5.3.10.7 Те ж саме для графа, у якому не менше чотирьох дерев.

5.3.10.8 Як шукати найкоротші шляхи у зваженому графі, якщо він є деревом?

5.3.11 Орієнтовані графи

5.3.11.1 Навести приклад орграфа і побудувати матрицю суміжностей.

5.3.11.2 Навести приклад оршляху, що не є орланцюгом.

5.3.11.3 Навести приклад орланцюга, що не є простим.

5.3.11.4 Навести приклад орциклу, що не є простим орциклом.

5.3.11.5 Навести приклад орграфа в якому усі вершини розташовані одночасно у двох різних орциклах.

5.3.11.6 Навести приклади орграфів з симетричною і несиметричною матрицями суміжностей.

5.3.11.7 Навести приклад орграфа, у якому кожна вершина має рівні вихідний і вхідний локальний степінь.

5.3.11.8 Навести приклад двох ізоморфних орграфів.

5.3.11.9 Навести приклад двох неізоморфних орграфів.

5.3.11.10 Навести приклади зв'язного і незв'язного орграфів.

5.3.12 Мережі і потоки в мережах

5.3.12.1 Навести приклад мережі і потоку у ній.

5.3.12.2 Навести мережу і функції на її ребрах, що не є потоком.

5.3.12.3 Навести приклад мережі і стаціонарного потоку у ній.

5.3.12.4 Навести приклад мережі і потоку у ній, що не є стаціонарним.

5.3.12.5 Навести приклад мережі і стаціонарного потоку у ній величиною три.

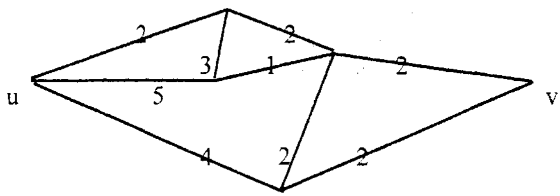
5.3.12.6 Навести приклад мережі і стаціонарного потоку у ній величиною три, що не буде максимальним.

5.3.12.7 Навести приклад мережі і стаціонарного потоку у ній величиною три, що не будуть максимальним.

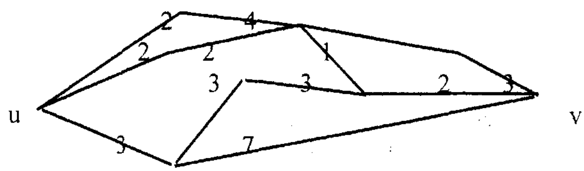
5.3.12.8 Чи можна побудувати приклад мережі і потоку у ній, що був би стаціонарним щодо двох пар «джерело-стік» і в обох випадках мав би ненульову величину?

5.3.12.9 Знайти максимальний стаціонарний потік з вершини u у вершину v для наведених нижче прикладів:

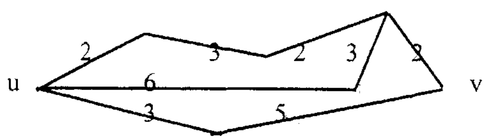
а)



б)



в)



5.4 Комбінаторика

5.4.1 Основні комбінаторні з'єднання

5.4.1.1 У ряду глядацького залу 15 крісел. Скількома способами можна розмістити на них 15 чоловік?

5.4.1.2 Скількома способами можна розфарбувати повний граф на 6 вершинах шістьма кольорами? (Два способи вважаються різними, якщо деяка вершина при одному способі має один колір, а при іншому способі – інший.)

5.4.1.3 Є квадратна матриця розміром 10х10. Вибираються з неї 10 елементів так, щоб ніякі два з них не належали однієї лінії. Скільки таких наборів по 10 елементів можна скласти?

5.4.1.4 Маємо десять співробітників і шість робочих місць, на які їх треба розподілити. Відомо, що кожен співробітник може працювати на кожному робочому місці. Скількома способами можна здійснити призначення?

5.4.1.5 На трьох комп'ютерах C_1, C_2, C_3, C_4, C_5 треба розв'язати три задачі P_1, P_2, P_3, причому кожна задача може розв'язуватися на будь-якому з комп'ютерів. Скількома способами можна направити задачі на розв'язання?

5.4.1.6 У дитячому саду є 5 дітей, яких треба розподілити в 7 груп, причому в одну групу не більше однієї дитини. Скільки є варіантів розподілу по групах?

5.4.1.7 Список екзаменаційних питань складається з 19 питань. З них потрібно скласти екзаменаційні квитки, причому в кожному квитку – рівно два питання. Скільки квитків можна скласти?

5.4.1.8 У виразі $(x+y)^{12}$ розкрили дужки і привели подібні члени. Який коефіцієнт буде стояти біля виразу x^4y^8?

5.4.1.9 Чому дорівнює сума $C_7^1 + C_7^3 + C_7^5 + C_7^7$?

5.4.1.10 Чому дорівнює сума $C_8^2 + C_8^4 + C_8^6 + C_8^8$?

5.4.1.11 У множині з 10 елементів зафіксовані чотири властивості p_1, p_2, p_3, p_4, які можуть мати, чи не мати елементи множини. Як за допомогою методу включення-виключення описати ті елементи, у яких немає жодного з даних властивостей?

5.4.1.12 Скільки відсутностей порядку існує на п'яти символах? На чотирьох символах?

5.4.1.13 Скільки зустрічей порядку 3 існує на п'яти символах? На шести символах?

5.4.2 Формальні степеневі ряди

5.4.2.1 Дана послідовність: $a_0 = 0$, $a_n = \frac{1}{n}$, $n = 1,2,\ldots$. Записати функцію, що її породжує.

5.4.2.2 Дана послідовність: $a_n = \frac{1}{n^2+1}$, $n = 0,1,2,\ldots$. Записати функцію, що її породжує.

5.4.2.3 Дано два формальних степеневих ряди:

$$f(t) = 1 + t + t^2 + \ldots + t^n + \ldots ;$$

$$g(t) = -1 + t - t^2 + t^3 - \ldots + (-1)^{n+1} t^n + \ldots .$$

Знайти $f(t) + g(t)$ і $f(t) - g(t)$.

5.4.2.4 Дані два формальних степеневих ряди:

$f(t) = 1 + t + t^2 + \ldots t^n + \ldots$ і $g(t) = 1 - t$.

Знайти добуток $f(t)g(t)$.

5.4.2.5 Знайти наступну частку від розподілу формальних степеневих рядів а) $\dfrac{1+t}{1-t}$; б) $\dfrac{1+t+t^2}{1-2t+t^2}$; в) $\dfrac{2t}{1-t}$.

5.4.2.6 Виконати арифметичні дії і записати формальний степеневий ряд у стандартній формі: $\dfrac{\dfrac{1-t}{2+t}}{(1+t+t^2+\ldots+t^n+\ldots)(2-t)}$.

5.4.3 Лінійні рекурентні співвідношення

5.4.3.1 Навести приклад трьох лінійних рекурентних співвідношень.

5.4.3.2 Розв'язати лінійні рекурентні співвідношення:

а) $a_n = 2a_{n-1}, n \geq 1, a_0 = 1$;

б) $a_n = a_{n-1} - a_{n-2}, n \geq 2,\ a_0 = 0, a_1 = 1$;

в) $a_n = 2a_{n-1} + a_{n-2}, n \geq 2, \quad a_0 = 1, a_1 = 1$;

г) $a_n = a_{n-1} + 2a_{n-2}, n \geq 2, \quad a_0 = 0, a_1 = 1$.

ЛІТЕРАТУРА

1. Гильберт Д., Бернойс П. Основания математики. – М.: Наука, 1979.
2. Горбатов В.А. Основы дискретной математики. – М.: Высш. школа, 1986.
3. Мальцев А.И. Алгебраические системы. – М.: Наука, 1970.
4. Мендельсон Н. Введение в математическую логику. – М.: Мир, 1974.
5. Яблонский С.В. Введение в дискретную математику. – М.: Наука, 1979.
6. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988.
7. Гаврилов С.П. Сапоженко А.А. Зборник задач по дискретной математике. – М.: Наука, 1978.
8. Нефедов В.Н., Осипова В.А. Курс дискретной математики. – М.: Издательство МАИ, 1992.
9. Кук Д., Бейз Г. Компьютерная математика. – М.: Наука, 1990.
10. Нефедов В.Н., Осипова В.А. Курс дискретной математики. – М.: Издательство МАИ. 1992.
11. Лекции по теории графов. / Емеличев В.А., Мельников О.И. и др. - М.: Наука, 1990.
12. Гаврилов С.П. Сапоженко А.А. Зборник задач по дискретной математике. – М.: Наука, 1978.
13. Бельский А.А. Теория графов и комбинаторика.: МИИТ,1979.
14. Харари Ф. Теория графов. - М.: Мир, 1998.
15. Басакер Р.,Саати Т. Конечные графы и сети. - М.: Наука, 1997.
16. Ахо А., Хопкрофт Дж.,Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир,1998.
17. Оре О. Теория графов. - М.: Наука, 1968.
18. Зыков А.А. Основы теории графов. - М.: Наука, 1987.
19. Риордан Дж. Комбинаторные тождества. - М.: Наука, 1982.
20. Кристофидес Н. Теория графов. Алгоритмический подход. - М.: Мир, 1997.
21. Комбинаторный анализ. Задачи и упражнения. Под ред. К. А. Рыбникова. - М.: Наука, 1982.

ЗМІСТ

Навчальне видання

Роїк Олександр Митрофанович
Тадевосян Роберт Геворкович

ОСНОВИ ДИСКРЕТНОЇ МАТЕМАТИКИ

ЧАСТИНА 2

ЕЛЕМЕНТИ ЗАГАЛЬНОЇ АЛГЕБРИ, БУЛЕВІ ФУНКЦІЇ, ТЕОРІЯ ГРАФІВ І КОМБІНАТОРИКА

Оригінал-макет підготовлено Роїком О.М.

Редактор С. А. Малішевська

Навчально-методичний відділ ВДТУ
Свідоцтво Держкомінформу України
серія ДК №746 від 25.12.2001
21021, м.Вінниця,Хмельницьке шосе, 95, ВДТУ

Підписано до друку 11.06.03р.
Формат 29,7х42
Друк різографічний
Тираж 85 прим.
Зам. № 2003-095

Гарнітура Times New Roman
Папір офсетний
Ум. друк. арк. 4,83

Віддруковано в комп'ютерному інформаційно-видавничому центрі
Вінницького державного технічного університету
Свідоцтво Держкомінформу України
серія ДК №746 від 25.12.2001
21021, м.Вінниця,Хмельницьке шосе, 95, ВДТУ